

UNIVERSIDAD PÚBLICA DE NAVARRA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INDUSTRIALES Y DE TELECOMUNICACIÓN



**"IMPLEMENTACIÓN DE UNA  
SOLUCIÓN PARA LA GESTIÓN DE  
UN CANAL DE TELEVISIÓN  
DIGITAL TERRESTRE"**

**Departamento de Ingeniería Eléctrica y Electrónica**

**Alumno:** Fernando Muñoz Fernández de Legaria  
Pedro Ansorena González

**Titulación:** Ingeniería de Telecomunicación

**Tutor:** Mikel Sagüés García

**Pamplona, 28 de Octubre de 2009**

*Gracias Mikel por todos los  
momentos que nos has dedicado  
en la creación de este trabajo,  
la mitad es gracias a ti :o)*

*Pedro y Fernando*

<b><u>UNIVERSIDAD PÚBLICA DE NAVARRA</u></b>	<b>- 0 -</b>
<b><u>ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN</u></b>	<b>- 0 -</b>
<b><u>1 INTRODUCCIÓN</u></b>	<b>- 6 -</b>
1.1 PROPÓSITO DEL DOCUMENTO	- 6 -
1.2 ANTECEDENTES Y OBJETIVO DEL PROYECTO	- 8 -
1.3 ESTADO DEL ARTE DE LA TELEVISIÓN DIGITAL INTERACTIVA EN NAVARRA	- 9 -
<b><u>2 INFRAESTRUCTURA NECESARIA PARA LA IMPLANTACIÓN DE UN CANAL LOCAL DE TELEVISIÓN DIGITAL INTERACTIVA EN EL VALLE DE ARANGUREN</u></b>	<b>- 13 -</b>
2.1 CONSIDERACIONES PREVIAS	- 13 -
2.2 DESCRIPCIÓN GENERAL DE LA INFRAESTRUCTURA	- 16 -
2.3 SOLUCIONES	- 20 -
2.3.1 SOLUCIÓN Nº 1: SOLUCIÓN PROFESIONAL CON MÁXIMA FLEXIBILIDAD Y CALIDAD DE COMPONENTES	- 21 -
2.3.2 SOLUCIÓN Nº 2: DIFERENTES SOLUCIONES QUE INTEGRAN VARIOS COMPONENTES EN DIFERENTES DISPOSITIVOS	- 35 -
2.3.3 SOLUCIÓN Nº 3: INYECCIÓN DE CONTENIDOS A PARTIR DE UN SERVIDOR DE APLICACIONES EMPLEANDO TECNOLOGÍA DE <i>STREAMING</i>	- 38 -
2.3.4 SOLUCIÓN Nº 4: <i>PLAYOUT</i> BÁSICO SIN APLICACIONES INTERACTIVAS NI GESTIÓN AVANZADA DE CONTENIDO	- 42 -
2.3.5 SOLUCIÓN Nº 5: DIGITALIZACIÓN DEL CANAL ANALÓGICO QUE SE TRANSMITE EN LA ACTUALIDAD	- 44 -
2.3.6 SOLUCIÓN Nº 6: CONTRATACIÓN DE UNA EMPRESA EXTERNA QUE PROPORCIONE TODOS LOS SERVICIOS Y SE ENCARGUE DEL MANTENIMIENTO	- 46 -
2.3.7 SOLUCIÓN Nº 7: INYECCIÓN DE CONTENIDOS CON LA AYUDA DE PROGRAMAS DE SOFTWARE LIBRE EMPLEANDO TECNOLOGÍA DE <i>STREAMING</i>	- 47 -
2.4 RECEPTORES	- 50 -
2.4.1 ADB-OSMOSYS	- 51 -
2.4.2 DIGIQUEST	- 53 -

2.4.3	ENGEL	- 53 -
2.4.4	HUMAX	- 54 -
2.4.5	I-CANT	- 54 -
2.4.6	INVEST	- 55 -
2.4.7	ILLUSION	- 55 -
2.4.8	STRONG	- 56 -
<b>2.5</b>	<b>RELACIÓN DE EMPRESAS Y DESCRIPCIÓN DE LAS SOLUCIONES APORTADAS POR LAS MISMAS A CADA UNA DE LAS PARTES DEL ESQUEMA GLOBAL</b>	<b>- 57 -</b>
2.5.1	ABE	- 57 -
2.5.2	ACTIVA MULTIMEDIA	- 58 -
2.5.3	ALCÁZAR DE SAN JUAN	- 61 -
2.5.4	APPEAR TV	- 64 -
2.5.5	CINEGY	- 65 -
2.5.6	DEKTEC	- 67 -
2.5.7	EUROTEK	- 69 -
2.5.8	GRASSVALEY THOMSON	- 70 -
2.5.9	ICAREUS	- 71 -
2.5.10	KRAMER ELECTRONICS	- 72 -
2.5.11	MAINCONCEPT	- 73 -
2.5.12	MIT-XPERS	- 73 -
2.5.13	OMB	- 74 -
2.5.14	PEBBLE BEACH	- 75 -
2.5.15	PROMAX	- 76 -
2.5.16	SAPEC	- 77 -
2.5.17	SIDSA	- 78 -
2.5.18	STREAM LABS	- 79 -
2.5.19	TANDBERG	- 80 -

2.5.20	TMIRA	- 81 -
2.5.21	TVMAKER (EPSILONMEDIA, VISION ROBOTICS)	- 83 -
2.5.22	VECTOR3 (VECTOR BOX)	- 84 -
2.5.23	VSN	- 85 -
<b>3</b>	<b><u>GESTIÓN DE UN CANAL DE TELEVISIÓN CON SOFTWARE LIBRE</u></b>	<b>- 88 -</b>
<b>3.1</b>	<b>OBJETIVO</b>	<b>- 88 -</b>
<b>4</b>	<b><u>GESTIÓN DE LA ESCALETA MEDIANTE EL VLC</u></b>	<b>- 90 -</b>
<b>4.1</b>	<b>VLC</b>	<b>- 91 -</b>
<b>4.2</b>	<b>VLM</b>	<b>- 96 -</b>
4.2.1	INTRODUCCIÓN COMANDOS VLM	- 96 -
4.2.2	¿QUE ES UN ARCHIVO DE CONFIGURACIÓN VLM?	- 96 -
4.2.3	COMANDOS VLM	- 100 -
4.2.4	USO DEL INTERFAZ WEB	- 102 -
<b>5</b>	<b><u>GESTIÓN DE LOS SERVICIOS DE TDT MEDIANTE OPENCASER</u></b>	<b>- 115 -</b>
<b>5.1</b>	<b>¿QUE ES OPENCASER?</b>	<b>- 115 -</b>
<b>5.2</b>	<b>INSTALACIONES NECESARIAS PARA EL USO DE OPENCASER</b>	<b>- 117 -</b>
5.2.1	INTRODUCCIÓN	- 117 -
5.2.2	INSTALACIÓN DEL SISTEMA OPERATIVO LINUX.	- 117 -
5.2.3	INSTALACIÓN DE OPENCASER Y LOS CONTROLADORES DE LA TARJETA DEKTEC	- 117 -
<b>5.3</b>	<b>USO DE OPENCASER</b>	<b>- 122 -</b>
5.3.1	INTRODUCCIÓN	- 122 -
5.3.2	DESCRIPCIÓN DEL SCRIPT USADO POR OPENCASER PARA LA CREACIÓN DE LOS TRANSPORT STREAM DE LAS TABLAS PSI-SI	- 125 -
<b>5.4</b>	<b>COMANDOS DEL OPENCASER</b>	<b>- 147 -</b>
5.4.1	INTRODUCCIÓN	- 147 -
5.4.2	ASPECTOS A TENER EN CUENTA AL USAR LOS COMANDOS DE OPENCASER	- 147 -

5.4.3	DESCRIPCIÓN DE LOS COMANDOS <i>OPENCATER</i> MÁS COMUNES A LA HORA DE EMITIR SERVICIOS DE TELEVISIÓN DIGITAL.	- 148 -
5.4.4	LISTADO DE TODOS LAS HERRAMIENTAS DISPONIBLES EN OPENCATER	- 152 -
<b>5.5</b>	<b>EMISIÓN DE UN TRANSPORT STREAM CON LA TARJETA DEKTEC</b>	<b>- 155 -</b>
5.5.1	PASOS PREVIOS A LA PUESTA EN EMISIÓN	- 156 -
5.5.2	<i>SCRIPT</i> DE COMANDOS OPENCATER PARA LA EMISIÓN DE UN SERVICIO DE TELEVISIÓN DIGITAL	- 157 -
5.5.3	<i>SCRIPT</i> DE COMANDOS OPENCATER PARA LA EMISIÓN DE DOS SERVICIOS DE TELEVISIÓN DIGITAL	- 163 -
<b>6</b>	<b><u>CODIFICACIÓN Y ADAPTACIONES DEL VÍDEO</u></b>	<b>- 172 -</b>
6.1	INTRODUCCIÓN	- 172 -
6.2	CODIFICACIÓN	- 173 -
6.3	ADAPTACIONES DEL VIDEO	- 177 -
6.3.1	INSERCIÓN EN LA ESCALETA Y EMISIÓN	- 177 -
6.3.2	RECOGIDA DEL VÍDEO: OPENCATER	- 178 -
<b>7</b>	<b><u>CONCLUSIONES Y RECOMENDACIÓN FINAL</u></b>	<b>- 181 -</b>

# 1 Introducción

## 1.1 Propósito del documento

Este documento se enmarca dentro del proyecto *OTRI 2009024051 "Propuesta de soluciones para la inserción de un canal de televisión propio del ayuntamiento en la red de cable del valle de Aranguren y de posibles aplicaciones de televisión digital interactiva a desarrollar"* suscrito entre el Ayuntamiento del Valle de Aranguren y la Universidad Pública de Navarra.

Dicho proyecto ha sido llevado a cabo por personal del Departamento de Ingeniería Eléctrica y Electrónica de la UPNA en colaboración con los Centros de Excelencia Software (CES) de Navarra [Ces1] que están integrados dentro del CEIN [Ce1].

A través de este documento se describen de manera detallada diferentes soluciones para la infraestructura necesaria a la hora de implantar un canal de televisión digital interactiva en el Valle de Aranguren.

Para ello, en el apartado 1.2 se comienza con un breve resumen de los antecedentes del proyecto y de los objetivos del mismo. Se continúa con una introducción al estado del arte de la televisión digital interactiva en Navarra, de tal forma que las soluciones propuestas puedan ser entendidas desde el punto de vista de la situación global en la que están enmarcadas (apartado 1.3).

En el capítulo 2 se entra en detalle en la descripción de la infraestructura así como en las diferentes soluciones a través de las cuales el objetivo final del proyecto puede llevarse a cabo.

Tras una breve introducción a los antecedentes de la red de cable del Valle de Aranguren, en el apartado 2.2. se describe la infraestructura genérica que es necesaria para la transmisión de un canal local de televisión digital, incluyendo la transmisión de aplicaciones interactivas. Dicha descripción genérica o global está dividida en módulos conceptuales de tal forma que es posible identificar las necesidades de la arquitectura desde un punto de vista sistémico.

Posteriormente, en el apartado 2.3 se concretan varias posibles soluciones basadas en instrumentos y componentes, software y hardware, existentes en el mercado para cubrir cada una de las necesidades del esquema global anteriormente mencionado. Dichas soluciones se detallan incluyendo un presupuesto para cada una de ellas y dejando las especificaciones técnicas de cada uno de los componentes para el apartado 2.5. Este apartado está ordenado en función de las empresas que proporcionarían las soluciones, describiéndose la funcionalidad de cada uno de los productos de las mismas que podrían ser empleados en cada una de las soluciones propuestas.

El apartado 2.4 está dedicado al estudio de los diferentes receptores de televisión digital interactiva que se pueden adquirir en el mercado a día de

hoy, haciendo especial hincapié en su capacidad para reproducir aplicaciones de televisión digital interactiva.

En el resto del documento, se trata de describir la puesta en marcha de la solución obtenida con software de libre distribución, la solución descrita en el apartado 2.3.7. El capítulo 3 se habla de este objetivo, de cómo poner en marcha un servicio de televisión digital con la mínima inversión posible. Una vez está claro el objetivo se pasa a comentar los programas que harán que este objetivo sea posible.

El capítulo 4 trata sobre como gestionar una escaleta de televisión con la aplicación *VídeoLAN Manager* dentro del software *VideolanClient*. Primero, se hace una breve introducción de software en concreto. Se comentan las utilidades del *VideolanClient*, de los protocolos que utiliza y de cómo se pueden aprovechar sus características para poner en marcha el canal de televisión digital.

Posteriormente, en el apartado 4.2 se entra más en concreto en la aplicación que se utiliza para hacer la automatización del canal. Primero se describen todos los comandos de que dispone, y seguidamente se pasa a describir la solución más adecuada para hacer uso de ellos.

El capítulo 5 está completamente dedicado a las herramientas denominadas como *Opencaster*. En primer lugar se trata de explicar que son estas herramientas y para que sirven. A continuación, en el apartado 5.2 se explica como obtener dicho software y como realizar una instalación completa del mismo.

En el resto del capítulo se explica como hacer uso de estas herramienta para poner en marcha esta solución. Se explican todas las herramientas de las que dispone *Opencaster*, entrando más en detalle en las que son indispensables para completar la solución en concreto. Además, en el apartado 5.3.2 se explica en detalle el *script* que utiliza *Opencaster* para crear un *Transport stream* con las tablas *PSI-SI*.

Para finalizar este capítulo, se describen los pasos a seguir para poner en funcionamiento una emisión de televisión digital.

En el último capítulo, el 6, se explica cómo realizar una codificación adecuada para la solución del apartado 2.3.7. Esta codificación se realizará mediante software de *Mainconcept*. Este software a pesar de ser de pago facilita en gran medida la codificación de los vídeos. Existen soluciones de libre distribución como el *ffmpeg*.

Para finalizar, se presentarán las conclusiones del presente documento, haciendo una recomendación final de la infraestructura a implantar en el Valle de Aranguren, sin perjuicio de que éste, a la luz de lo descrito en este documento, pueda optar por cualquier otra de las soluciones.



## **1.2 Antecedentes y objetivo del proyecto**

El Valle de Aranguren está formado por las poblaciones de Mutilva Baja, Mutilva Alta, Tajonar, Zolina, Labiano, Góngora, Ilundain, Laquidain y Aranguren está situado en la zona sur de Pamplona, tiene una superficie de 45 km<sup>2</sup> y, tal y como destaca en su propia página web, es conocido como el pulmón verde de la Comarca de Pamplona debido a su riqueza paisajística e itinerarios peatonales.

Además de esta perspectiva ligeramente bucólica del valle, la realidad es que el Ayuntamiento del Valle de Aranguren destaca también por una apuesta clara en pos de la modernización de todos los servicios proporcionados a sus ciudadanos. En este sentido, es uno de los ayuntamientos pioneros en Navarra en diferentes iniciativas de carácter tecnológico, como pueden ser una red propia de televisión por cable propia, una web con información municipal actualizada, la oferta de internet a sus ciudadanos por un precio que casi puede considerarse simbólico, la introducción de la tarjeta ciudadana y otros muchos.

Enmarcado en esta tendencia por parte del Ayuntamiento de ofrecer servicios a sus ciudadanos se encuentra un nuevo proyecto de generación y transmisión de contenidos audiovisuales propios a través de la red de cable del valle. Para ello, el Ayuntamiento ha procedido a la compra de la instrumentación necesaria para la grabación y edición de productos audiovisuales, entre los que se encuentran dos cámaras de grabación de vídeo digital de alta definición, una plataforma de edición de vídeo, así como otros elementos necesarios, como puede ser la habilitación de un local para la realización de los mencionados productos audiovisuales.

Otra de las posibilidades que abre el mundo de la televisión digital es la transmisión de aplicaciones interactivas que se ejecutan en los receptores de los usuarios pudiendo de este modo ofrecer diferentes servicios. Hasta el momento son pocos los Ayuntamientos, al menos en España, que han apostado por este tipo de aplicaciones como una plataforma de comunicación con sus ciudadanos. Sin embargo, las grandes posibilidades abiertas por esta tecnología no han pasado desapercibidas para los responsables de Ayuntamiento del Valle de Aranguren.

A raíz de todo lo anterior, el Ayuntamiento del Valle de Aranguren acudió a la Universidad Pública de Navarra con el objetivo de conocer la forma en que dichos productos audiovisuales (incluyendo las aplicaciones interactivas) podían ser transmitidos a través de su red de cable, proponiéndole su participación a modo de consultoría en el diseño de la infraestructura necesaria para llevar a cabo dicho proceso.

Como consecuencia de dicho acercamiento y de una serie de reuniones para concretar el ámbito y alcance de dicha colaboración, se firmó un contrato de tres meses entre el Ayuntamiento del Valle de Aranguren, los Centros de Excelencia Software de Navarra y la UPNA, cuyos objetivos fundamentales se resumen en los siguientes puntos:

- Proponer varias implementaciones para la infraestructura del canal de televisión propio, siendo estas soluciones de diferentes precios y funcionalidades.
- Hacer una propuesta de posibles contenidos para aplicaciones MHP a desarrollar en un futuro.
- Buscar subvenciones para la implantación del canal de TV (Gobierno de Navarra, Ministerio de Industria Turismo y Comercio, planes Avanza, etc.).

El canal de televisión que se pretende implantar es de carácter público, entendiéndose como un servicio del Ayuntamiento hacia sus ciudadanos, por lo que el objetivo es mostrar información municipal (cultura, agenda, noticias, etc.) y contenido de video/audio producido por el propio valle. Por lo tanto, y dado lo costoso en tiempo y material humano de este tipo de producciones, esto implica que la producción audiovisual será relativamente escasa.

También es importante destacar que las aplicaciones de televisión digital interactiva que se pretende proponer para un futuro desarrollo irán de más sencillas a más complejas (con presupuestos separados) de tal forma que el Valle de Aranguren pueda decidir aquellas en las que esté mas interesado, o contratar un desarrollo de las mismas de manera escalonada en el tiempo. Las aplicaciones de televisión digital interactiva pueden dividirse en dos grupos: informativas e interactivas. Las primeras no requieren de un canal de retorno, mientras que las segundas si lo necesitarán. Dado que la consultoría a realizar debe contemplar diferentes posibilidades, se presentarán soluciones que permiten la utilización del canal de retorno y otras que no lo harán.

En el presente documento se describe la infraestructura necesaria para la implementación del canal de televisión, mientras que las aplicaciones interactivas propuestas para la inclusión en el mismo se describen en un segundo documento [Ces2].

### ***1.3 Estado del arte de la televisión digital interactiva en Navarra***

La televisión evoluciona dejando atrás las técnicas analógicas para digitalizarse y ofrecer al usuario mayor calidad y nuevos servicios. La televisión a día de hoy, es el medio de comunicación de mayor penetración entre la población española, cubriendo el 90% de los ciudadanos. Esto unido a las aproximadamente 4 horas que pasa de media un telespectador al día, nos da una visión de lo fácil y atractivo que es llegar al ciudadano a través de la Televisión Digital Interactiva.

Con la nueva tecnología digital se pueden entremezclar contenidos audiovisuales tradicionales con uno o varios canales de datos, posibilitando la transmisión de aplicaciones software multimedia que se ejecutarán en un televisor digital o en un televisor analógico dotado de un decodificador o *Set Top Box*. Estas aplicaciones pueden sincronizarse con contenidos tradicionales y proporcionar interactividad con el usuario.

El estándar *Multimedia Home Platform* (MHP) desarrollado por DVB es el *middleware* abierto que permite desarrollar aplicaciones interactivas permitiendo al usuario final disfrutar de un nuevo concepto de televisión [Dvb1, Mhp1]. A diferencia de la televisión actual meramente pasiva, la nueva TDT se convierte en un sistema bidireccional en que el espectador puede interactuar con los servicios y aplicaciones que recibe, todo ello a través del *Set Top Box*.

Con la inminente llegada de la televisión digital, la oferta digital está sobre la mesa, y el estándar *MHP* (*Multimedia Home Platform*) se convierte de facto en la especificación a seguir por todos los proveedores de contenidos de la televisión digital terrestre (TDT), aunque diferentes movimientos en la industria podrían hacer peligrar este escenario.

El inminente apagón analógico que tendrá lugar en Navarra el 31 de diciembre del presente año, hace que tanto los usuarios como los distribuidores tengan que acelerar el cambio hacia la televisión digital. A pesar de que gracias a tener una red de cable propia el Valle de Aranguren podría seguir emitiendo en analógico sin ningún problema, parece más que evidente que el propio "empujón legislativo" debe ser aprovechado para realizar una transición que, por otra parte, será inevitable a corto plazo. Además, dado que los canales de televisión digital terrestre se introducen de manera transparente en la red de cable de Valle de Aranguren, todo lo referente a dicho modo de transmisión afectará directamente a los vecinos de valle y, por lo tanto, debe ser tenido en cuenta.

Sin embargo, el panorama nacional en lo referente al apagón analógico es bastante más complejo de lo que en un primer momento se puede entender. En particular, se podría decir que estamos asistiendo a la primera de una serie de transiciones hacia la televisión digital y, como se verá a continuación, cada una de estas transiciones va a tener un impacto sobre el usuario, que se verá obligado a actualizar su receptor para seguir el ritmo de la tecnología.

De este modo, podemos dividir el proceso de transición en televisión digital (y en particular en televisión digital terrestre) en varias fases:

- **Apagón analógico:** Cese de las emisiones analógica de televisión al que nos hemos referido anteriormente. Su implicación no va más allá del hecho de que todos los receptores de televisión analógicos deben ser sustituidos por televisores con capacidad para sintonizar señales digitales, o en su defecto, será necesaria la introducción de un *Set Top Box* para realizar esta conversión de analógico a digital.

Esta es la primera transición, la que tendrá lugar a finales de este año, y para la que la mayoría de los usuarios se están preparando mediante la compra de *Set Top Boxes* de muy bajo coste (alrededor de 30 euros) o mediante la compra de televisores con pantalla plana que, en su mayoría, ya incorporan la capacidad de recibir señales digitales.

- **Acceso condicional:** En agosto de este año, y de manera bastante sorprendente por otra parte, El Consejo de Ministros ha aprobado hoy un Real Decreto-Ley que regula la prestación del servicio de televisión digital terrestre (TDT) de pago mediante descodificador para las concesiones de ámbito estatal.

La normativa establece que los sistemas de acceso condicional (descodificadores) que se utilicen en la futura TDT de pago deberán basarse en un estándar tecnológico abierto, de forma que cualquier aparato pueda servir para descodificar la señal de pago de todos los operadores que utilicen este tipo de emisión. Sin embargo, este último punto es de difícil cumplimiento, ya que no se puede limitar la compra por parte de los usuarios de un equipo que no cumpla con estas condiciones.

Evidentemente, esto puede considerarse como una segunda transición, ya que la inmensa mayoría de los decodificadores que se habían vendido hasta la fecha, y los que previsiblemente se van a seguir vendiendo hasta Navidad, no incorporan ningún sistema de acceso condicional. Además, los que sí lo incorporan no tienen porque utilizar el mismo sistema de acceso condicional que alguna de las cadenas que desde la aprobación del decreto ley ya emiten contenidos codificados.

- **Alta definición:** Aunque a día de hoy a nivel nacional solamente se están realizando algunas pruebas piloto en este sentido, la televisión digital terrestre de alta definición es una tecnología plenamente madura y que ya está en explotación en varios países de nuestro entorno, como Italia y Noruega. Así mismo, empleando tecnología satelital, a día de hoy ya es posible recibir emisiones en alta definición a nivel nacional, a través de algunas plataformas de pago.

Este hecho, junto con el indudable mercado que se habrá en retransmisiones deportivas debido al gran salto de calidad que experimenta el usuario final al visualizarlas de este modo, hace inevitable que en breve las cadenas comiencen a emitir contenidos en alta definición.

Llegado ese momento (que a buen seguro veremos a lo largo de 2010), los decodificadores estándar, o los decodificadores con acceso condicional que se hayan vendido hasta la fecha quedarán obsoletos. De hecho, incluso las televisiones de pantalla plana que la mayoría de los usuarios compran pensando que le proporcionan la posibilidad de decodificar contenidos en alta definición, no están realmente preparadas para ello. Por poner un ejemplo de esto último, las televisiones con el sello "HD Ready" son capaces de reproducir contenidos en alta definición, pero necesitan de un decodificador externo para la recepción y decodificación de los mismos (más detalles sobre este tema se pueden encontrar en la referencia [Min1]).

En definitiva, en un plazo relativamente corto, los receptores adquiridos hasta la fecha volverán a quedar obsoletos.

- **Aplicaciones MHP:** Por último, pero no menos importante, está el tema de las aplicaciones de televisión digital interactiva. La ley obliga a los distribuidores de televisión digital terrestre a reservar una parte muy importante de cada uno de sus canales para la emisión de contenidos interactivos. A día de hoy se realizan emisiones de aplicaciones interactivas, pero la inmensa mayoría de los ciudadanos no tiene un receptor preparado para la recepción de las mismas. Además, la mayoría de las aplicaciones que se emiten tienen un interés relativamente bajo para los ciudadanos. Sin embargo, este ancho de banda tiene que ser ocupado de alguna forma, y no cabe duda de que las aplicaciones que despierten el interés de los usuarios terminarán por aparecer. En particular, estas aplicaciones tienen un especial interés en canales locales, ya que pueden ser empleadas como nuevos canales de información municipal o, gracias al canal de retorno, como una vía para que los ciudadanos se comuniquen con su administración.

En definitiva, los receptores de televisión digital deberían estar capacitados para la recepción y reproducción de aplicaciones MHP, y la mayoría de los que ahora se venden en el mercado no lo están.

Uno de los objetivos del trabajo realizado a cargo del presente proyecto ha sido el intentar dar con una tecnología que permitiera a los vecinos del valle de Aranguren afrontar las transiciones a las que se ha hecho mención más arriba sin tener que hacer ningún cambio en sus equipos domésticos. De este modo, se pretendía encontrar un receptor que estuviera dotado, además de la inevitable decodificación de señales de televisión digital estándar, de un sistema universal de acceso condicional, de la capacidad para decodificar señales digitales de alta definición así como de trabajar con aplicaciones de televisión digital interactiva MHP de última generación (MHP 1.1.3).

Sin embargo, a la conclusión que se ha llegado, y que se detalla en el apartado 2.4 de este documento, es que dicho receptor de televisión digital no existe a día de hoy. O al menos, no existe destinado al mercado español.

## **2 Infraestructura necesaria para la implantación de un canal local de televisión digital interactiva en el Valle de Aranguren**

### **2.1 Consideraciones previas**

La red de telecomunicaciones del valle de Aranguren se configura como una red de cable con arquitectura tipo híbrido fibra coaxial. En la memoria de la auditoría de red realizada a raíz del Proyecto OTRI 2006 12 128, "Red de telecomunicación valle de Aranguren" [Ara1], se puede encontrar una descripción minuciosa de las características generales de este tipo de redes. La red provee actualmente servicios de televisión (canales libres) y acceso Internet en el área de Mutilva Alta y Mutilva Baja.

La red esta diseñada en una estructura en árbol, como se puede observar en la figura 1, y no esta dotada de ningún tipo de redundancia. Toda la red cuelga de un mismo elemento: la cabecera; es decir, todos los nodos y troncales coaxiales están directamente conectados con la cabecera, ya sea por medio de fibra óptica, ya sea por medio de cable coaxial.

Es una red pequeña, en la que no se pueden distinguir todos los elementos de una red HFC convencional debido a su reducido tamaño.

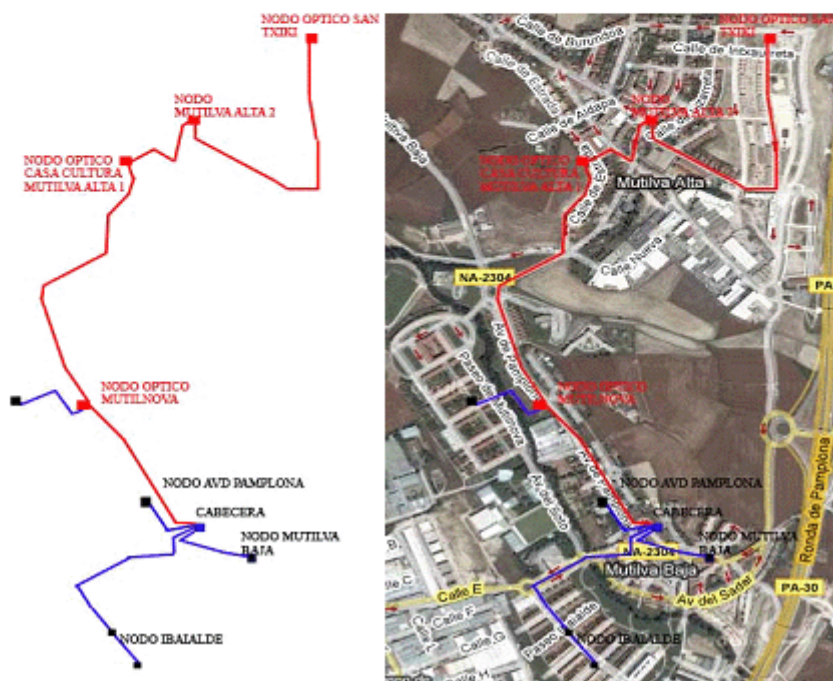
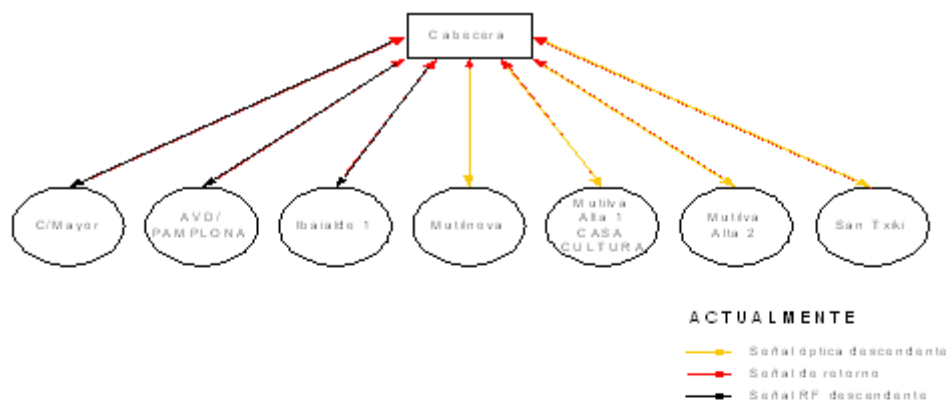
Los elementos que conforman esta red son:

- El sistema de captación.
- La cabecera (que puede considerarse un nodo primario).
- La red de distribución óptica.
- Nodos finales (PTRO).
- Ramificaciones troncales coaxiales.
- La red de distribución coaxial.

El primer elemento que aparece es el sistema de cabecera, desde el que se captan las señales que se van a distribuir a lo largo de la red.

En la cabecera se produce el procesamiento de las señales obtenidas en el sistema de cabecera. Todas las señales sufren una reconversión en frecuencia para alojarse en la parte baja del espectro de RF. Además, las señales de satélite digital y de satélite analógico se transmodulan para distribuirse en formato TV-AM (PAL B/G). Las únicas señales de televisión que no se transmiten moduladas en amplitud son las de TDT (Televisión Digital Terrestre). Es evidente que con la venida del apagón analógico todos los canales que a día de hoy son modulados en amplitud, deberán ser digitalizados para su correcta recepción en los hogares del valle. En cualquier caso, el modo o los medios por los que esta digitalización se lleve a cabo quedan al margen de los objetivos del presente proyecto.





**Figura 1. Diagrama de la red HFC del valle de Aranguren**

Además de producirse el procesamiento de las señales, es en la cabecera donde se inserta la señal de Internet. Para ello se utiliza un canal de 8 MHz, más concretamente el canal S35. La señal descendente de Internet se genera en el CMTS (*CableModem Termination System*), que es el dispositivo encargado de gestionar todos los enlaces de datos (*EuroDocsis*), tanto para el enlace descendente, en el que comparte un canal de TV para su transmisión a toda la red, como para el canal ascendente, en el que se tienen cuatro enlaces dedicados, uno para cada zona de la red. La salida del único puerto de señal descendente del CMTS se conecta a un combinador, donde se mezcla con las señales de TDT. Este combinador se conecta con un repartidor, en el cual se mezclan todas las señales de la cabecera. Ésta se combina con las señales de TV y se amplifica para llevar a cabo su distribución por la red.

Para una descripción detallada de la arquitectura de la red actual, así como de la posible evolución de la misma, se puede consultar el documento "Desarrollo y evolución de la red de cable de Aranguren" [Ara1].

En lo referente al presente proyecto, las dos conclusiones fundamentales que se pueden obtener a partir del estudio de la red de cable del Valle de Aranguren son las siguientes:

- El Ayuntamiento del Valle de Aranguren dispone de una red con una excelente capacidad para la distribución de televisión digital, en la cual es posible la inclusión de canales modulados en COFDM (estándar de modulación empleado en televisión digital terrestre), tal y como se demuestra al insertar directamente los canales provenientes por este medio de transmisión.
- La red de telecomunicaciones anteriormente mencionada dispone de un canal de retorno de mediana capacidad, pero que es suficiente para dar servicio a posibles aplicaciones de televisión digital interactiva que se desee transmitir por la red de cable y que requieran de un canal de retorno. A la hora de diseñar las aplicaciones de televisión digital interactiva a introducir en la red de cable, deberá tenerse en cuenta las limitaciones derivadas de este canal de retorno. Este es un punto fundamental, ya que no parece lógico obligar a los ciudadanos que ya disfrutan del servicio de internet proporcionado por el Ayuntamiento a tener que contratar otro servicio en paralelo para poder hacer uso de las aplicaciones de televisión digital interactiva.



## 2.2 Descripción general de la infraestructura

En el presente apartado, se comienza por describir la arquitectura genérica de una red de televisión digital, desde la grabación de los contenidos mediante cámaras digitales, hasta su reproducción en los receptores de los usuarios finales, pasando por todos los mecanismos de gestión tanto del contenido audiovisual como de las aplicaciones interactivas que se pueden insertar en dicha red. Posteriormente, en el apartado 2.3 de este informe se detallarán las soluciones particulares por las cuales este esquema general puede ser llevado a la práctica.

La infraestructura necesaria para la implantación de un canal de televisión digital interactiva se compone de diferentes elementos que interaccionan entre si tal y como se describe en la figura 2. En la figura los nombres aparecen en inglés, por ser esta la terminología más habitual empleada por todas las empresas del sector. De este modo, se pretende que los términos a los que se hace referencia sean lo más universales posibles, lo que, sin duda, ayudará a la hora de identificarlos en las especificaciones dadas por cada una de las empresas. En los casos en que la terminología anglosajona sea muy similar a la castellana, se utilizará esta última forma.

A continuación se describe de manera somera la función de cada uno de estos elementos en dicho esquema global:

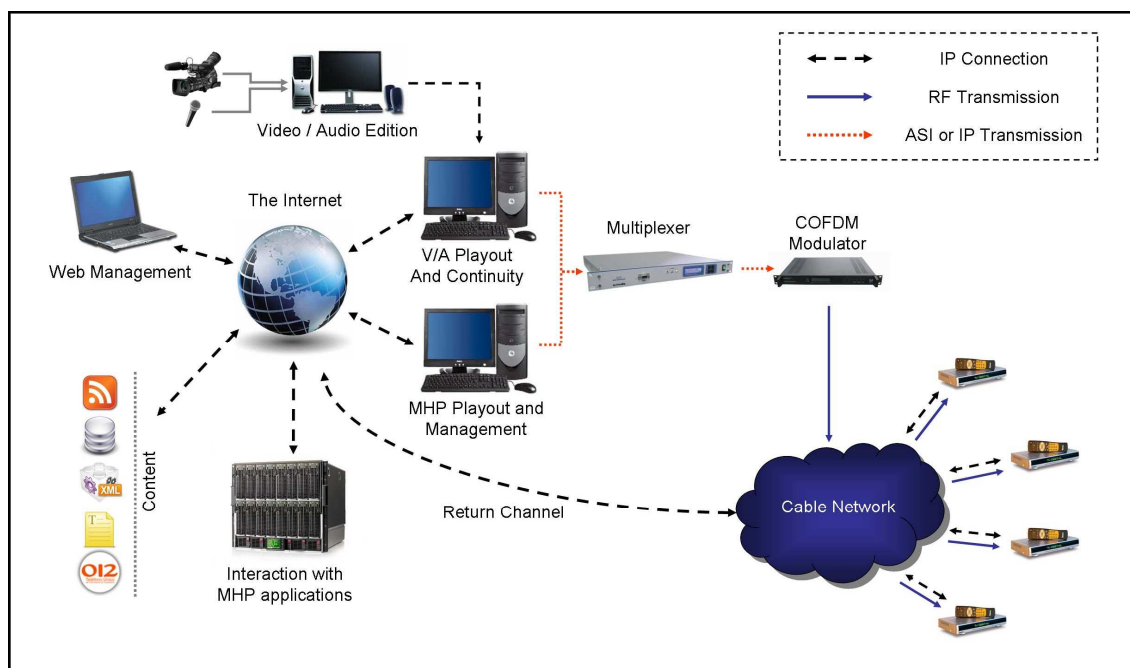
- **Internet:** A día de hoy, Internet es uno de los elementos clave en cualquier tecnología. En este caso también, Internet supondrá el nodo global a través del cual muchos de los elementos del canal de televisión serán integrados. En la figura 2 se muestra de manera gráfica dicha función.
- **Gestión Web:** ("Web Management", en el esquema) De manera ideal, la gestión y control de todos los procesos y sistema involucrados debería hacerse a través de una interfaz web. En la mayoría de las soluciones propuestas en el apartado 2.3, esto será posible en su totalidad o para alguna de las partes del sistema.
- **Producción de vídeo y audio:** ("Video/Audio Edition", en el esquema) Incluye toda la infraestructura necesaria para la producción de contenidos audiovisuales, desde la captura de la imagen y el sonido hasta su almacenamiento en disco duro o publicación en directo.
- **Continuidad y Playout de Vídeo y Audio:** ("V/A Playout and Continuity", en el esquema) Transmisión de la trama digital con los contenidos audiovisuales a retransmitir sobre la red. Dentro de este apartado se incluirá todo el sistema de gestión de los contenidos audiovisuales generados en la fase de producción. En definitiva se trata del bloque conceptual encargado del almacenamiento y gestión de toda la producción audiovisual generada por el Valle de Aranguren. La forma en que el contenido audiovisual es transmitido dependerá de la solución particular que se adopte.
- **Gestión y Playout de las aplicaciones interactivas:** ("MHP Playout and Management", en el esquema) Gestión y transmisión de las aplicaciones interactivas a insertar en el canal local. Uno de los

objetivos prioritarios del presente proyecto era la capacitación del canal de televisión local del Valle de Aranguren para la transmisión de aplicaciones interactivas. Este es el bloque responsable de la gestión y transmisión de dichas aplicaciones, es decir, el almacenamiento, el control de el calendario de emisión de dichas aplicaciones, su señalización en la red... etc.

- **Contenido:** ("Content", en el esquema) Este bloque del esquema global se refiere a los contenidos o servicios que se pretende ofrecer a la ciudadanía del Valle de Aranguren a través de las aplicaciones interactivas. La localización de estos contenidos puede ser muy variada, desde servidores de internet remotos, hasta contenidos que se insertan de modo manual en las propias aplicaciones interactivas. En un segundo documento adjunto a este se detallan tanto las aplicaciones que podrían transmitirse a través de la red de cable, como la ubicación de los contenidos que puedan requerir dichas aplicaciones [Ces2]. En definitiva, este punto queda fuera del ámbito de este documento.
- **Interacción con las aplicaciones interactivas:** ("*Interaction with MHP applications*", en el esquema) En caso de que el contenido mostrado por las aplicaciones interactivas se encuentre en un sitio remoto (un servidor web, por ejemplo) la interacción entre las aplicaciones y dicho contenido debe hacerse a través de una "pasarela" cuyo objetivo es que el acceso al contenido sea transparente para las aplicaciones, en relación a la fuente desde donde se encuentra esta información. La implementación concreta de esta "pasarela" depende fundamentalmente de la solución adoptada para la programación de las aplicaciones interactivas, por lo que también queda fuera del alcance de este documento.
- **Multiplexación:** ("*Multiplexer*", en el esquema) En este bloque conceptual se produce la fusión entre los contenidos audiovisuales las aplicaciones de televisión digital interactiva. Como norma general, aunque no tiene porque realizarse de esta forma en todos los casos, se trata de la multiplexación de la trama de transporte de vídeo y audio con la trama de transporte de aplicaciones interactivas. Su objetivo es generar una única trama de transporte en formato *DVB-ASI* que englobe todos los contenidos a transmitir en el canal de televisión del Valle de Aranguren.
- **Modulación:** ("*Modulation*", en el esquema) La señal digital en formato *DVB-ASI* a la salida del multiplexador debe de ser modulada con el objetivo de poder ser transmitida por la red de cable del Valle de Aranguren. En este punto, es conveniente hacer notar que el esquema de modulación que se ha elegido para la red de cable es el especificado por el estándar *DVB-T* [Dvb2], el cual emplea la modulación *COFDM* (*Coded Orthogonal Frequency Domain Modulation*). Este es el esquema de modulación utilizado en televisión digital terrestre por lo que, en principio, no parece el más adecuado para la transmisión en redes de cable. Sin embargo, la robustez de este sistema de modulación lo habilita para su empleo en redes de cable, tal y como ha sido demostrado en una gran cantidad de municipios y pequeñas urbanizaciones. Además, la elección de este esquema de modulación permitirá el empleo de receptores convencionales de televisión digital terrestre en los hogares de los

ciudadanos del Valle de Aranguren, rebajando de esta forma el precio de dichos receptores por ser estos los más extendidos a día de hoy en el mercado nacional.

- **Red de cable y receptores de televisión:** ("Cable Network", en el esquema) Se trata de la red de cable del Valle de Aranguren, y de los receptores de televisión digital interactiva instalados en los hogares de los vecinos del valle. Tal y como se ha descrito en el apartado 2.2, la red de distribución no necesita ser modificada para la inserción del canal propio.



**Figura 2. Esquema global de la infraestructura a desplegar.**

Además de los elementos fundamentales que se han descrito, también se ha tenido en cuenta prestaciones adicionales de la red y que, en un principio, no fueron incluidas en las especificaciones del proyecto. En cualquier caso, estas serán siempre tratadas como características opcionales:

- Posibilidad de insertar servicios propios del Ayuntamiento con contenidos audiovisuales de alta definición (HD). Dado que el equipo de grabación adquirido en su día por el Ayuntamiento del Valle de Aranguren, está preparado para la captura de vídeo en alta definición, la posibilidad de transmitir dicho contenido sin reescalarlo a resolución estándar (SD) también ha sido contemplada en varias de las soluciones.
- También se ha trabajado en la posibilidad de transmitir en directo contenidos propios del Ayuntamiento. Nótese que en los objetivos iniciales del proyecto solamente se contemplaba la posibilidad de transmitir contenido propio del Ayuntamiento previamente editado y almacenado en disco duro. Sin embargo, en algunas de las soluciones propuestas será posible la emisión de vídeo en directo, por lo que

esta funcionalidad también ha sido tomada en cuenta a la hora de evaluar cada una de ellas.

## 2.3 Soluciones

En este apartado se describen las diferentes soluciones que se podrían adoptar para la implantación de un canal de televisión digital propio en el valle de Aranguren. Las soluciones se describen de mayor a menor funcionalidad, contemplando en cada caso los presupuestos para cada una de ellas.

A lo largo de este apartado, la descripción de las diferentes soluciones se realiza sin detenerse constantemente a detallar características técnicas de los componentes que se utilizan en las mismas. En el apartado 2.5 se introduce brevemente cada una de las empresas cuyos productos son mencionados en el presente apartado y se proporciona una breve descripción de las características técnicas los mismos.

Además, en todos los casos se hace referencia a las especificaciones técnicas de estos productos, las cuales se pueden encontrar en un CD adjunto al documento.

Por otro lado, en la descripción de las soluciones no se tendrá en cuenta los receptores que se utilicen en la vivienda del usuario, ya que estos son independientes de la solución. En su lugar, las diferentes opciones disponibles en el mercado de receptores se detallan en el apartado 2.4 de este documento.

Tampoco se hará hincapié en el hecho de disponer de Internet, en la parte del sistema dedicada a la interacción con las aplicaciones ni en los propios contenidos a los que las aplicaciones necesitan acceder. Todas estas cuestiones se dan por soportadas por el sistema.

Finalmente, dado que las cámaras de grabación y la plataforma de edición de vídeo ya han sido adquiridas por el ayuntamiento y, de hecho, ya se están produciendo contenidos, tampoco se entrará a estudiar diferentes soluciones para este bloque. En concreto, el material de producción audiovisual disponible en el Valle de Aranguren a día de hoy es el siguiente:

- Vídeo
  - Cámara de vídeo *Sony* HDV/DV HVR-V1P (x2)
    - Ya montado en cámara: *eyecup* grande, batería recargable, parasol con tapa, protector de viento y micrófono.
  - Trípode con rotula *Manfrotto* mba g80p (x2)
  - Foco cámara de video *Vitan* con cargador
- Estación de edición
  - CPU
  - monitores TFT 19" *Hyundai* x90w LCD
  - 1 monitor 17" Visualización *Hyundai*
  - *Avid liquid*
  - Modem *motorola* s85101e
  - Tarjeta de video *Dazzle DV Editor*
  - *Windows Microsoft Office*

- Audio
  - Mesa de sonido *Yamaha* mg 206 c
  - Micrófono Sh58 Legendary Vocal Microfone
  - Micrófono de petaca Sennheiser w 300
  - Pie de micro Xym König & Meyer Mikrofonstativ
- Almacenamiento
  - Memoria externa *Lacie* 1TBytes
  - Memoria externa *Toshiba* 250 Gbytes

Por lo tanto, se considera que se dispone de unos vídeos almacenados en un ordenador (en el formato que se desee) siendo irrelevante la forma en que han sido capturados y editados.

En definitiva, este apartado se centra en:

- El modo en que las **aplicaciones interactivas** pueden ser gestionadas y transmitidas.
- La manera en que se transmite y gestiona todo el **material audiovisual**, almacenado o en directo, que se desea transmitir.
- La **multiplexación** de ambos contenidos (aplicaciones MHP y contenido audiovisual)
- La adaptación del conjunto de servicios a transmitir para su transmisión por la red de cable (**modulación**).

A pesar de que las soluciones que se describen son independientes entre sí, se recomienda leer el documento secuencialmente, ya que muchos de las soluciones tienen partes en común y, para evitar una excesiva redundancia, solamente se discuten en la primera ocasión que se da la cuestión.

### 2.3.1 Solución Nº 1: Solución profesional con máxima flexibilidad y calidad de componentes

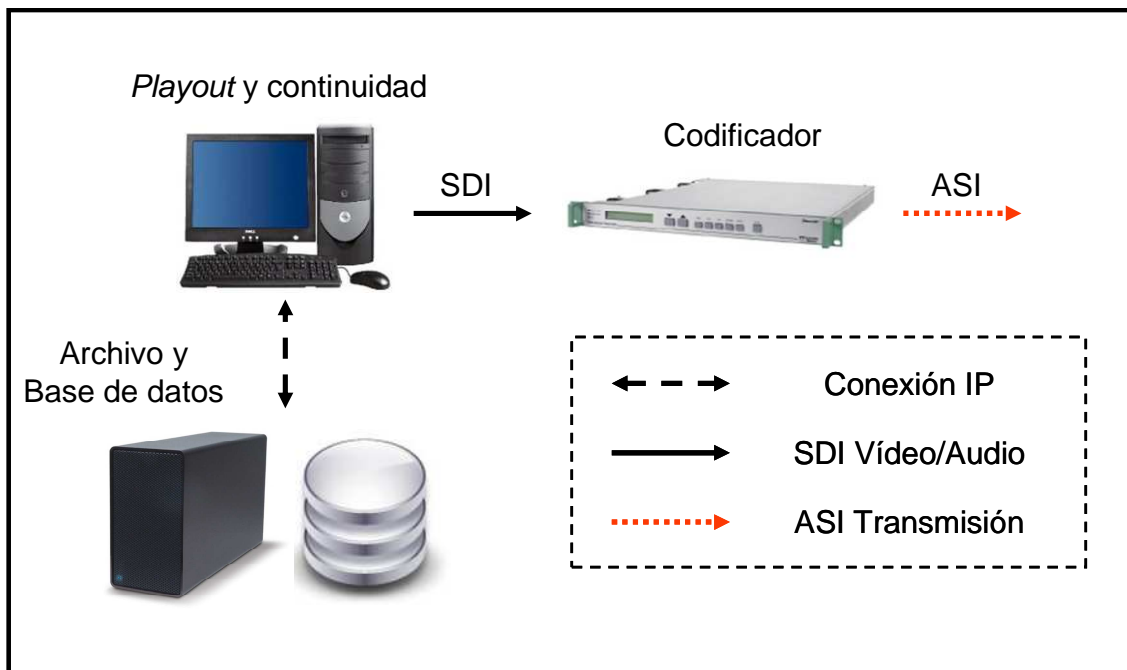
En esta sección se describe la que podría ser denominada “la solución profesional” para disponer de un canal de televisión digital propio. Esto es así, ya que los instrumentos que se emplean en la misma, y las interfaces a través de las que se interconectan los mismos, son los que se usan de manera sistemática en todas las televisiones de grandes dimensiones.

El esquema global, por lo tanto, se corresponde exactamente con el de la Figura 2, siendo todos los elementos presentes en la misma componentes independientes, o incluso estando alguno de estos bloques dividido en más de un componente.

En particular, es necesario detallar en mayor medida la manera en que se construye el bloque del *playout* de vídeo.

#### Continuidad y *Playout* de vídeo y audio

La figura 3 muestra en mayor detalle los elementos que componen este bloque:



**Figura 3. Detalle del sistema de continuidad, gestión y playout de vídeo.**

Como se puede observar en la figura, el *playout* está compuesto por varios elementos. En primer lugar es necesario un ordenador que será el encargado de transmitir la señal digital con el contenido de audio y vídeo. Además, debe ser capaz de garantizar en todo momento la continuidad de esta emisión. Para ello, dispondrá de un software que en la industria viene a llamarse "continuidad", y que es el que permite la realización y control de la parrilla y escaleta de emisión del canal de televisión. Entre otras, las funcionalidades básicas de este software son:

- Controlar la emisión de contenidos, garantizando que se emitan en el instante en que estaban programados.
- Insertar logotipos, textos o pequeñas animaciones superponiéndolos al vídeo en emisión (este proceso puede recibir el nombre de "titulación").
- Control de los dispositivos a partir de los cuales se desea recibir el vídeo a transmitir. En la figura 3 se muestra un disco duro y una base de datos, pero el vídeo a transmitir podría provenir de diferentes discos duros, de reproductores de vídeo o VTR-s (*Video Tape Recorder*, en terminología anglosajona) en distintos formatos (VHS, DVD, DV... etc.). En todos los casos, será el software y hardware de continuidad el encargado de controlar dichos dispositivos.
- Ingesta de contenidos. La ingesta consiste en la adaptación de los distintos formatos de codificación de vídeo y audio en los que pueda estar codificado un material audiovisual, para poder ser almacenados, gestionados y transmitidos por el software de continuidad. En definitiva, se trata de convertir el contenido audiovisual a la entrada del mismo en un contenido apto para trabajar con él.

Como regla general, la señal a la salida de la plataforma de continuidad estará codificada siguiendo el estándar SDI (*Serial Digital Interface*, en



terminología inglesa) [Sdi1]. El estándar SDI es un estándar de codificación de vídeo digital pensado para la comunicación de equipos dentro de un estudio de televisión. La calidad del vídeo codificado de esta forma es máxima, ya que se trata de un vídeo sin ninguna compresión, exceptuando el submuestreo de color. Además, el estándar SDI permite embeber el vídeo con el audio en una única trama de datos digitales, lo que hace que la totalidad de la información proveniente de las diferentes cámaras (u otras fuentes de vídeo) viaje por un único cable.

La mayoría de las plataformas de continuidad de vídeo trabajan con este estándar. Sin embargo, un vídeo en formato SDI no sirve para las dos únicas cosas que una cadena de televisión quiere hacer con un vídeo: almacenarlo y transmitirlo. Esto es debido a que el volumen de datos generado por un vídeo sin compresión es extremadamente alto, lo que lo hace físicamente inviable para esos dos cometidos. Para tener un orden de magnitud, un vídeo codificado en SDI tiene una tasa de transferencia de datos de aproximadamente 270 Mbps. Esto implica que en un DVD cabría escasamente un minuto y medio de vídeo en este formato.

Por lo tanto, se hace indispensable la compresión de ese vídeo antes de poder transmitirlo. Es en este punto donde entra a trabajar el codificador que se muestra en la figura 2. Su función será, por lo tanto:

- Comprimir el vídeo en formato SDI a su entrada, para permitir su transmisión por una red de datos. En una cadena de televisión en Europa (la norma es diferente en otros países) esta compresión se hace siguiendo las recomendaciones del consorcio DVB [Dvb1], el cual define dos estándares para la transmisión de vídeo:
  - Para transmisión de vídeo con resolución estándar (SD, *Standar Definition*, en inglés) se impone la utilización del estándar *MPEG-2*. Tras pasar por el codificador *MPEG-2* el vídeo en resolución estándar tendrá una tasa de transferencia de 3 a 5 Mbps. Además, en lo que se refiere a la calidad percibida por el espectador, esta drástica reducción en su *bitrate* es prácticamente imperceptible para el ojo humano, siempre que la codificación se haya realizado correctamente.
  - Para transmisión de vídeo en alta resolución (HD, *High Definition*, en inglés) se recomienda usar el estándar *MPEG-4* en lugar del *MPEG-2*, debido a su mayor coeficiente de compresión. Evidentemente, esta mayor eficiencia de compresión será a costa de una mayor complejidad en el hardware y software de los receptores de televisión digital, pero es la única forma de hacer viable una transmisión en alta definición.
- La segunda función de un codificador es adaptar la señal de vídeo para su transmisión por una red de televisión y su posterior recepción en un receptor de televisión digital estándar. Esta codificación implica la inserción de una serie de datos de información y servicio, así como la paquetización de la trama de datos. En este proceso, cada uno de los elementos fundamentales del contenido a transmitir se introduce en una trama elemental de datos (ES, *Elementary Stream*, en inglés). De este modo, si el contenido a transmitir está compuesto por un



vídeo y dos audios (uno de ellos, por ejemplo, en versión original), estos se dividirían en tres tramas elementales. Del mismo modo, la señalización a introducir en la trama también se divide en tramas elementales. Finalmente, todas las tramas elementales que conforman los datos a transmitir, son multiplexadas en una única trama final o trama de transporte (TS, *Transport stream*, en inglés).

Los detalles de los dos procesos anteriores quedan totalmente fuera del alcance de esta memoria, pero en resumen se podría decir que a la salida del codificador el vídeo digital SDI ha sido transformado en una trama de transporte con una tasa de transferencia de datos y una codificación que lo hacen apto para su transmisión por una red de televisión que cumpla con el estándar DVB. Como norma general, la salida del codificador se dará a través de una interfaz ASI (*Asynchronous Serial Interface*, en terminología inglesa). En algunos casos se suele emplear la terminología *ASI-DVB* o *DVB-ASI*, para referirse a una trama de datos codificada de esta forma.

En definitiva, esta trama de datos ya estará preparada para ser introducida en un modulador que la adapte para su transmisión por una red de televisión.

Anteriormente hemos mencionado brevemente el tercero de los componentes de la figura 2, el de "Archivo y base de datos". En su versión más sencilla, se trata de un disco duro que podría estar instalado en la misma máquina donde se aloja el software de continuidad.

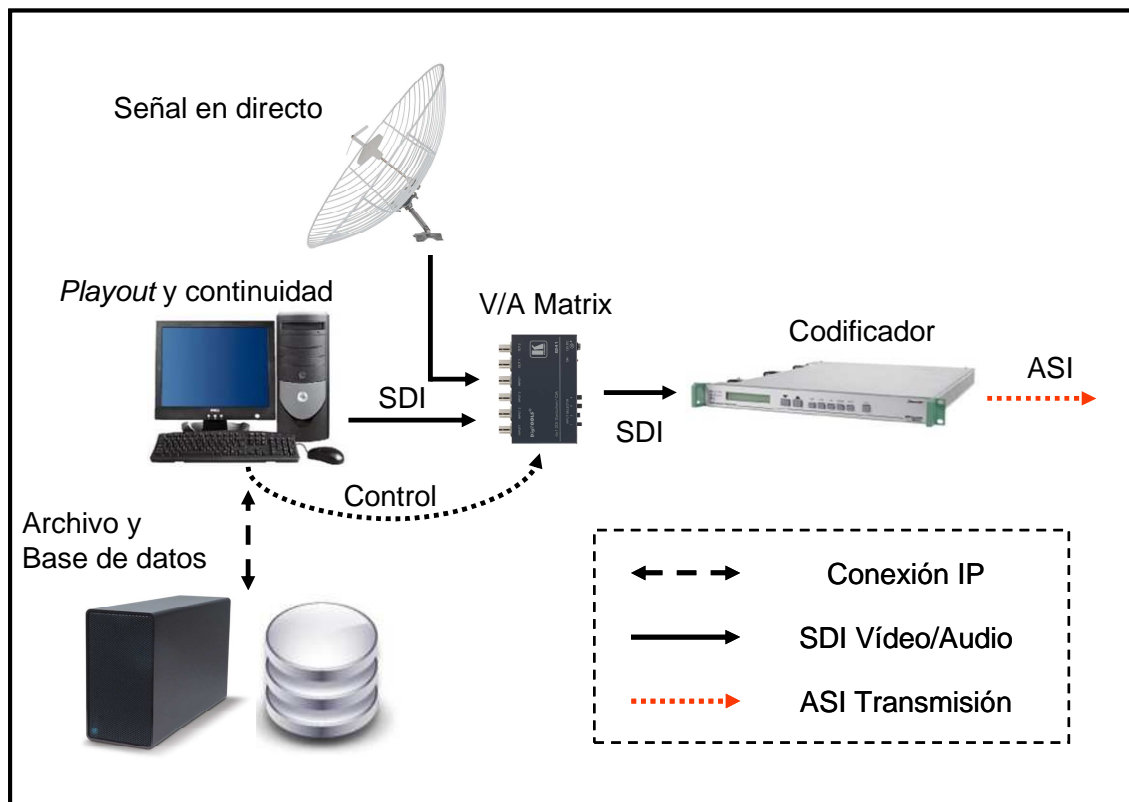
Sin embargo, su alcance es mucho más extenso en una plataforma de televisión profesional. La organización y gestión de todo el material audiovisual generado por una cadena de televisión es un problema realmente complejo y que hay que manejar con mucho cuidado. Para ello, las plataformas de *playout* y continuidad de vídeo suelen disponer de programas específicos para el almacenamiento del material y su organización en bases de datos. De este modo, cada vez que un material audiovisual es ingestado, se "obliga" al usuario que maneja el programa a introducir una serie de datos en los formularios de esta base de datos. De este modo la localización del vídeo se simplifica sobremanera.

Otro aspecto a tener en cuenta es el tamaño y las propiedades de los discos duros en los que se almacena el contenido. Los habrá de mayor o menor tamaño, de mayor o menor velocidad de acceso a datos y de mayor o menor precio. Todos esos parámetros son tenidos en cuenta en una cadena de televisión, de tal forma que los materiales menos utilizados se almacenen en discos duros lentos, de gran capacidad y baratos, mientras que los contenidos a los que es necesario acceder de manera inmediata se almacenan en discos de alta velocidad.

En definitiva, se trata de una utilidad de la plataforma de *playout* y continuidad que facilita la gestión de todo el contenido audiovisual. La facilidad con que los diferentes puestos de edición que componen una cadena de televisión, la potencia del software de edición que incorporan, la comodidad de las herramientas de comunicación que proporcionan o, por

supuesto, el precio de todas estas utilidades, son otros de los parámetros clave a la hora de elegir una plataforma de gestión.

Otro aspecto que no se ha tenido en cuenta hasta el momento, es la posibilidad de emitir contenido audiovisual en directo. Para ello, el esquema mostrado en la figura 2 debe ser completado, tal y como aparece en la figura 3.



**Figura 4. Detalle del sistema de continuidad, gestión y playout de vídeo.**

La parabólica de la figura 4 representa cualquier fuente de vídeo, en formato SDI, que se desee transmitir en directo. En el caso de que lo que se desee transmitir sea la propia señal de las cámaras de vídeo adquiridas por el ayuntamiento, y dado que estas no disponen de un puerto SDI, sería necesario emplear un adaptador de vídeo. En cualquier otro caso, como también ocurre en el caso de una señal proveniente de una emisión satelital, la señal deberá ser decodificada y trasformada a SDI. Dicha señal será introducida en una matriz de conmutación de vídeo. Una matriz de conmutación de vídeo SDI dispone de varias entradas y varias salidas que podrán ser controladas por la plataforma de *playout* de tal modo que esta controle cual de sus entradas es transmitida hacia el codificador.

La introducción de una matriz de conmutación y de una plataforma de *playout* con capacidad para controlarla puede dar una gran flexibilidad a una cadena de televisión, ya que le permite conmutar entre varias fuentes de vídeo diferentes (los VTR-s, DVD's, reproductores VHS y demás componentes que se mencionaban anteriormente estarán conectados a dicha matriz).

## Gestión y *layout* de las aplicaciones interactiva

Otro de los grandes bloques del esquema propuesto en la figura 2 es el de "Gestión y *layout* de las aplicaciones interactivas". Al igual que en el caso de los contenidos audiovisuales, las aplicaciones interactivas también han de ser codificadas en una trama de transporte para poder ser transmitidas por la red. En este sentido, hay que resaltar que el estándar DVB permite la transmisión de canales de televisión cuyo contenido sean únicamente aplicaciones interactivas, por lo que la señalización y paquetización a la salida de este bloque tiene que cumplir con todas las propiedades definidas por la norma.

Por lo tanto, la salida de este bloque también será una trama de transporte que se transmite a través de una interfaz *DVB-ASI*.

Una aplicación de televisión digital interactiva esta conformada por una serie de ficheros que se deben ejecutar en el receptor del usuario. En definitiva, es exactamente lo mismo que una aplicación software que se ejecuta en un ordenador convencional. Probablemente la mayor diferencia entre ambos mundos estriben en que en un ordenador personal se pueden ejecutar aplicaciones en muchos formatos diferentes (java, flash, .net, word, excel... etc) mientras que en un receptor de televisión digital las aplicaciones tienen que ser siempre del mismo tipo. En concreto, el consorcio DVB define las aplicaciones MHP [Mhp1] como el tipo de aplicaciones a ejecutar en un receptor de televisión digital interactiva. Los receptores, por lo tanto, deberán tener instalado un *middleware* MHP para poder ejecutar dichas aplicaciones. Un *middleware* puede definirse en pocas palabras como la capa del receptor que se encarga de comunicar a las aplicaciones con el hardware de la máquina, para que estas puedan realizar sus funciones. Las aplicaciones MHP están escritas en Java siguiendo una API abierta. Esto, y el hecho de ser las recomendadas por el consorcio DVB, hace que a día de hoy pocos fabricantes duden en relación al *middleware* a instalar en un receptor interactivo. En el apartado 2.4 y en el documento relativo a las aplicaciones a desarrollar para el Valle de Aranguren [Ces2] se pueden encontrar más detalles sobre este tema.

Por lo tanto, el objetivo fundamental de este bloque es el de preparar las aplicaciones para poder ser transmitidas por la red. Sin embargo, existen otra serie de cuestiones a tener en cuenta a la hora de decantarse por una u otra plataforma de transmisión y gestión de aplicaciones:

- La manejabilidad y sencillez de la interfaz gráfica a través de la que se gestiona la plataforma.
- El alcance de la señalización de información y servicio en la trama de transporte emitida. Este es un parámetro muy importante, ya que algunas plataformas no son capaces de introducir una señalización completa y requieren de un multiplexor a la salida para completar esta información.
- La capacidad de la plataforma para funcionar con aplicaciones MHP avanzadas así como para gestionarlas (generación de *stream events*, programación de la escaleta de aplicaciones, capacidad y

funcionalidad de su base de datos, posibilidad de remultiplexar tramas de transporte a la entrada de la plataforma... etc.)

## Multiplexador y modulador

Estos son los dos elementos más sencillos desde el punto de vista conceptual de toda la cadena, aunque su precio suele ser bastante elevado.

Un multiplexador DVB es un instrumento que dispone de varias entradas *DVB-ASI* y una o más salidas *DVB-ASI*. Su función es la de multiplexar tramas de transporte que cumplen con el estándar DVB, transformándolas en una única trama de transporte. Los parámetros fundamentales de un multiplexador son:

- Estabilidad y fiabilidad. Este instrumento tiene que funcionar de manera continua durante años, siendo además el punto donde se juntan varios caminos de una cadena de producción de televisión digital. Por lo tanto, la fiabilidad del sistema debe ser máxima.
- Manejabilidad de su interfaz de control. Muchos multiplexadores en el mercado carecen de una interfaz gráfica, por lo que tienen que ser gestionados a través de línea de comandos o a través de su panel frontal. En cualquier caso, en una televisión pequeña (como es el caso de Aranguren) este hecho pierde importancia, ya que será extraño que la red necesite ser reconfigurada a menudo.
- Flexibilidad o capacidad de la multiplexación. Con esto se hace referencia a la flexibilidad que otorga el multiplexador para construir la trama de transporte a su salida. En muchos casos la señalización de diferentes flujos elementales a la entrada del multiplexor será la misma, lo que, de no poder modificar dicha señalización, haría que la trama de transporte a su salida fuera incorrecta. Además, otros parámetros como su capacidad para reconocer y modificar la señalización de las aplicaciones interactivas son críticos en muchos casos.

Por su parte, el modulador es el elemento que se encarga de adaptar la trama de transporte para que pueda ser transmitida por una red de televisión. Tal y como se ha comentado con anterioridad, en el caso de la red de cable del valle de Aranguren se recomienda modular la señal en el mismo formato empleado por la televisión digital terrestre, con el objetivo de:

- Reducir el coste de los receptores y del propio modulador, por ser este el formato de modulación más empleado a día de hoy.
- Ser compatible con los canales de televisión digital terrestre que a día de hoy se están emitiendo en la red de cable del Valle de Aranguren. Estos canales son recibidos a partir de antenas terrestres y tras un filtrado y amplificación son introducidos en la red, sin modificar su formato de modulación. En caso de elegir una modulación diferente para el canal propio del ayuntamiento, esto obligaría a los usuarios de la red a utilizar dos receptores.

En definitiva, en la propuesta actual el modulador recibirá una trama de transporte a través de un puerto *DVB-ASI* y sacará una señal modulada en COFDM siguiendo las especificaciones del estándar *DVB-T*.

### Presupuestos y comentarios sobre componentes

La tabla 1 muestra las diferentes opciones que se han encontrado en la industria para cada uno de los bloques a los que se ha hecho referencia anteriormente. La solución de menor precio se compone, evidentemente, seleccionando el componente de menor precio de cada apartado, mientras que la "solución de calidad óptima" es aquella que los redactores de este documento recomiendan en caso de que el Valle de Aranguren optara por una solución completa como esta. En esta y en sucesivas tablas de presupuestos, la opción recomendada para cada bloque del presupuesto se señala con el siguiente símbolo: €<sup>R</sup>.

En algunos casos se añade en la parte inferior del precio las palabras "+ PC", las cuales indican que dicho componente debe ser instalado en un ordenador personal convencional, y que el precio de dicha máquina no se ha considerado en el presupuesto.

Por último, comentar que al lado del nombre de los componentes se ha escrito su referencia bibliográfica, con el objetivo de facilitar la consulta de sus especificaciones completas y el presupuesto proporcionado por la empresa.

Concepto	Precio
Plataformas de gestión, almacenamiento y <i>playout</i> de contenidos	
Stream Labs, <b>Wind SDI II</b> [StrP1]	<b>2.100 €</b> <b>+ PC</b>
Vector3, <b>Vectorbox 6000</b> [VecP2]	<b>26.610 €</b>
Vector3, <b>Vectorbox DSX</b> [VecP1]	<b>16.240 €</b>
VSN, <b>vsnOneTV</b> [VsnP1]	<b>16.500 €<sup>R</sup></b>
Codificadores <i>MPEG-2</i>	
Saptec, <b>Sivac 3000 MVE3010-D4A</b> [SapP1]	<b>3.961 €<sup>R</sup></b>
Tandberg, <b>E5710 MPEG-2 SD Encoder</b> [TanP1]	<b>14.571 €</b>
Matriz de conmutación SDI (opcional)	
Kramer, <b>VS-41HD</b> [KraP1]	<b>2.338 €</b>
Servidor de aplicaciones MHP	
Grassvalley, <b>Coral MHP Server</b> [GraP1]	<b>18.000 €</b>
Icareus, <b>PC100 Compact</b> [IcaP1]	<b>13.300 €</b>
MIT Xperts, <b>iMUX Multiplexer</b> [MitP1]	<b>9.251 €<sup>R</sup></b>
Tmira, <b>TmBroadcast</b> [TimP1]	<b>35.000 €</b>
Multiplexadores DVB	

Abe, <b>MUX 1002</b> [AbeP3]	<b>4.550 €<sup>R</sup></b>
AppearTV, <b>MC 3000 Multiplexer</b> [AppP1]	<b>8.000 €</b>
Dektec, <b>DTA-105 + DTC-700 MuxXpert + DTU-225</b> [DekP6-7]	<b>3.085 € + PC</b>
Eurotek, <b>EK-PWS/11 + EK-AMX/4</b> [EurP2]	<b>5.200 €</b>
Grassvaley, <b>Amber II Starter</b> [GraP1]	<b>7.500 €</b>
Saptec, <b>MTS1000-D4A</b> [SapP1]	<b>7.747 €</b>
Moduladores COFDM	
OMB, <b>Modulador DVB-T/H</b> [OmbP2]	<b>13.113 €</b>
Promax, <b>Modulador DVB-T MO-160</b> [ProP2]	<b>4.000 €</b>
Promax (bastidor), <b>DT-800 + DT-900 + DT-101 + DT-710 + DT-901</b> [ProP1]	<b>3.042 €<sup>R</sup></b>
Dektec, <b>DTA-112-SP</b> [DekP3]	<b>1.295 € + PC</b>
<b>Total (menor precio):</b>	<b>22.904 € + PC</b>
<b>Total (Solución calidad óptima<sup>R</sup>):</b>	<b>37.304 €*</b>

**Tabla 1. Presupuesto Solución N° 1 (precios sin IVA).**

Por último, se realizan una serie de comentarios sobre los diferentes componentes, justificando en cada caso la elección recomendada. Sin embargo, se recomienda una lectura del apartado 2.5 en donde se resumen las características de todos los componentes, así como de las hojas de especificaciones de los diferentes componentes.

## **Playout**

En relación a el *playout*. Las empresas *VSN* y *Vector3* son indudablemente las referencias del mercado nacional en lo referente a las plataformas de continuidad. Hay muchas otras, algunas de las cuales pueden consultarse en el apartado 2.5, pero dadas las excelentes prestaciones proporcionadas por *VSN* y *Vector3*, y dado que el mantenimiento, compra y posterior actualización (en caso de que se considere necesario) resulta más sencillo, son estas las que se proponen.

Tanto *VSN* como *Vector3* integran dentro de su oferta diferentes funcionalidades añadidas al simple *playout* de los contenidos de vídeo. Por un lado, disponen de un excelente software de continuidad con funcionalidades avanzadas, tanto en lo referente al manejo de la aplicación como a la inserción de textos o logotipos. Además, el software viene instalado en una máquina para *rack* (el precio, por lo tanto, incluye el

---

\* La solución recomendada no incluye la matriz de conmutación SDI, por ser este un elemento que puede ser incorporado al sistema posteriormente sin ninguna complejidad añadida.



hardware de la máquina) que ya dispone de discos duros de alta velocidad, conectados a través de puertos SCSI. Del mismo modo, incorporan un módulo de ingesta profesional y módulos de control de dispositivos externos, como matrices de vídeo, reproductores, etc.

En definitiva, se trata de empresas con una gran experiencia en el sector y que ofrecen un producto de primerísima calidad a un precio razonable. En particular, el servicio otorgado por VSN es excelente, según varias cadenas de televisión a las que se ha consultado, siendo su plataforma extremadamente escalable y flexible. Por lo tanto, en caso de que el Ayuntamiento de Aranguren opte por una solución profesional como esta, VSN es la recomendación de los redactores de este documento.

En caso de que no se desee adoptar una solución de esta envergadura, la alternativa propuesta es el software de continuidad **TeleInfo** de la casa **Stream Labs** el cual incluye una tarjeta **Wind SDI II**. Esta es una solución implantada en varias televisiones a lo largo del mundo, pero sus prestaciones son considerablemente menores que las de las anteriores. En primer lugar, la tarjeta no viene instalada en una máquina específica, sino que se instala en un ordenador convencional. Esto que a priori es una ventaja normalmente terminará convirtiéndose en una desventaja debido a la menor fiabilidad de la máquina. Además, la plataforma no dispone de módulos de ingesta avanzados, ni es capaz de controlar dispositivos externos. Sin embargo, si dispone de una entrada SDI, por lo que sería capaz de intercalar el contenido propio almacenado en disco duro con otros contenidos en directo (adaptados a SDI).

Por lo tanto, se trata de una solución perfectamente válida, pero con menores prestaciones y una menor escalabilidad de cara a un futuro. Sin embargo, dadas la pequeña magnitud del canal local que se pretende implantar, esta podría ser suficiente. De hecho, será la que se recomiende en el resto de las soluciones que requieran una plataforma de continuidad con salida SDI. En todas ellas, este producto podrá ser “conmutado” por el de VSN o Vector3.

## Codificador

En lo relativo al codificador se presentan únicamente dos opciones a pesar de que, como es evidente, en el mercado existen muchas más. En concreto, se trata del codificador **Sivac 3000 MVE3010-D4A** de la casa **Saptec** y del codificador **E5710 MPEG-2 SD Encoder** de la casa **Tandberg**. El primero es el codificador más barato que se ha podido encontrar (con unas prestaciones mínimas, por supuesto) mientras que el segundo es uno de los mejores codificadores del mercado. Desde un punto de vista de sus especificaciones, ambos son tremendamente similares, pero la fiabilidad, flexibilidad y calidad de compresión proporcionada por el producto **Tandberg** es mucho mayor. Sin embargo, en la mayoría de los casos este salto de calidad no va a ser percibida por el usuario y dados los requerimientos del canal de televisión a implantar en el Valle de Aranguren el codificador **Saptec** es más que suficiente y es el recomendado por los autores de este documento. Resaltar que **Saptec** no es una pequeña empresa que esté despuntando, sino que suministra equipos a grandes televisiones a nivel

nacional e internacional. En definitiva, sus equipos han sido testeados y están en funcionamiento por lo que ofrecen totales garantías.

## Servidor de aplicaciones MHP

En el apartado de los servidores de aplicaciones MHP se presentan cuatro opciones. La más cara de todas ellas, *Tmira*, ha sido incluida por ser esta una empresa con sede en Cuenca y en Madrid a diferencia del resto cuyas sedes son extranjeras. En definitiva, *Tmira* proporcionará una atención mucho más cercana que el resto por ser, además, una empresa relativamente pequeña. La funcionalidad de sus productos es muy elevada, estando por encima de las otras opciones en casi todos los parámetros. Sin embargo, es una opción tremendamente cara, por lo que solo se recomienda en caso de que se pretenda subcontratar el desarrollo de las aplicaciones interactivas a esta misma empresa, en cuyo caso, el desarrollo de las mismas sería realmente rápido.

*Icareus* también podría ser una buena opción, pero por diversos problemas su plataforma no ha podido ser testada, lo que nos deja con la duda de su verdadera funcionalidad. Esta empresa se incluye, ya que dispone de un servidor de vídeo con funcionalidad muy limitada, pero que se integra de manera sencilla con su servidor de aplicaciones, lo que permitiría disponer de una solución completa a cargo de una única empresa y por un precio razonable (18.000 €, además de un modulador). Esta podría ser una buena opción, por el hecho de que simplifica sobremanera las actualizaciones y el mantenimiento.

*Grassvaley Thomson* es una de las mayores empresas en el sector de la televisión digital a nivel mundial, por lo que sus productos son de una calidad excelente. Sin embargo, su presupuesto es relativamente alto por lo que no se recomienda su adquisición.

La recomendada por lo autores de este documento es adquirir el servidor de la casa *MIT xperis* debido a su precio relativamente asequible (de los mejores del mercado) y a que tanto la UPNA como los CES ya han trabajado con esta solución con anterioridad y la conocen de primera mano. Es una solución flexible que permite realizar de manera sencilla la inserción de las aplicaciones, el control de la programación de su edición y que, además, soporta funcionalidades de alto nivel como la inserción de *Stream Events*. Además, en los CES se dispone de la herramienta de autor de esta misma casa, lo que podría facilitar el desarrollo de las aplicaciones interactivas para el Valle de Aranguren, en caso de que esta segunda fase se pusiera en marcha. Por último, resaltar que esta herramienta se utiliza en varias de las mayores compañías dedicadas a la interactividad en televisión a nivel nacional, como son *Iecisa*, *Net2U* o *MaatG*. En definitiva, una solución con un excelente precio, buenas prestaciones y testada en multitud de empresas del sector.

## Multiplexores



Los multiplexores son, como ya se ha comentado, un componente fundamental del sistema. Existen mucho, aunque no se ha encontrado uno por el que los autores se decanten de manera rotunda. Por un lado, la mayoría de los fabricantes son extranjeros, y por el otro las soluciones más baratas no ofrecen la fiabilidad necesaria.

La recomendación, por lo tanto, es adquirir el multiplexor de la casa *ABE*, el cual está escalado a las características del proyecto (dos entradas ASI y una única salida ASI) y es un instrumento de garantías. El único pero es que se trata de una empresa italiana y que no dispone de un distribuidor en España. Por lo tanto, el Ayuntamiento del Valle de Aranguren deberá estudiar la idoneidad de adquirir este producto, ya que en ocasiones los problemas administrativos podrían suponer un coste mayor que la adquisición de otro producto ligeramente más caro.

La solución basada en la empresa *Dektec* está basada en la integración en un ordenador personal de dos tarjetas de vídeo y un software de multiplexación. Es una solución viable y flexible, ya que permitiría realizar actualizaciones de este software para añadir nuevas funcionalidades o incluso la inserción de nuevas tarjetas para realizar otras funciones (como la captación y remultiplexación de otras fuentes de vídeo, o incluso la emisión de modo rudimentario de contenido propio almacenado en el ordenador). Su coste, además, es el más bajo de todas las soluciones aportadas, aunque el hecho de requerir de un PC (de prestaciones considerablemente altas) hace que a nivel económico esté a la altura o por encima del multiplexor *ABE*. Además, la estabilidad y fiabilidad del sistema será menor que en el resto, además de ocupar mucho más espacio. Por lo tanto, en principio, tampoco se recomienda esta implementación.

La opción de *Eurotek* es muy flexible, por tratarse de una solución modular basada en Linux, pero desde el punto de vista de la impresión subjetiva adquirida de la empresa no resulta demasiado fiable.

El resto de opciones para este apartado son soluciones profesionales de grandes empresas del sector. El multiplexador *Saptec* no es demasiado caro, y al ser de la misma empresa que el codificador recomendado su integración sería sencilla. Lo mismo podría decirse del multiplexador *Grassvaley* en relación a su servidor MHP. Por último, *AppearTV* dispone de un producto excelente y que fue recomendado por los fabricantes del *MIT Xperts iMux*.

En resumen, desde un punto de vista económico la recomendación de los autores es apostar por el multiplexor de la casa *ABE* en caso de que esto no suponga ningún problema administrativo para el ayuntamiento. En caso de que esto si sea un problema, lo idóneo sería optar por el multiplexor *Saptec*, facilitando de este modo su integración con el codificador, además de disponer de un servicio mucho más cercano por ser esta una empresa con sede en Madrid. En caso de que el presupuesto sea un punto crítico y de no poder ser adquirida la solución de *ABE*, se podría trabajar con la basada en productos *Dektec* (tiene una excelente red de distribución en España), pero a sabiendas de la fiabilidad de esta solución.

## Moduladores DVB-T

En el apartado de los moduladores se debe hacer una primera puntualización o puesta en situación. El Ayuntamiento del Valle de Aranguren a día de hoy dispone de una cabecera de televisión digital, la cual recibe las señales procedentes de emisiones satelitales o terrestres y las adapta para su transmisión por la red de cable. A día de hoy esta adaptación consiste en un cambio de frecuencia de las emisiones analógicas para su transmisión por la red de cable y una transmodulación de las señales procedentes de satélite a banda lateral vestigial (formato de modulación utilizado en televisión analógica).

A partir del 31 de diciembre, esta cabecera deberá ser adaptada, ya que cesarán las emisiones analógicas terrestres y los usuarios adaptarán sus equipos domésticos para la recepción de señales digitales. En definitiva, la transmodulación que a día de hoy se hace de los canales satelitales dejará de tener sentido, teniendo que modificarse varios equipos en la cabecera para modularlas en formato digital.

En principio, este proceso es independiente del trabajo realizado a cargo del presente proyecto, pero la solución adoptada en el apartado de modulación si que podría verse influida por la solución que se adopte en dicha adaptación.

Por lo general, las señales captadas en la cabecera son de baja potencia, por lo que cada uno de los canales analógicos que se transmiten a día de hoy a de ser filtrado y amplificado empleando un amplificador monocanal. Sin embargo, cuando se genera un canal propio, es posible alimentar la red con una potencia considerable gracias a equipos de modulación específicos para este cometido. Por lo tanto, existen dos aproximaciones a la modulación de un canal propio:

- Modularlo con baja potencia, e incorporarlo a la red como si fuera otro cualquiera de los canales que se están captando.
- Modularlo con alta potencia y mezclarlo con otro grupo de canales en uno de los combinadores que a día de hoy existen en la cabecera.

Como es evidente, en el segundo caso se requiere un equipo de mejores prestaciones que en el primero. Otro aspecto evidente a partir de lo anterior es que la puesta en marcha del canal local deberá hacerse en colaboración con la empresa que adapte la cabecera.

En la tabla 2 se han propuesto varias soluciones para ambos casos. Las dos primeras, el modulador de la casa *OMB* y el modulador **Modulador DVB-T MO-160** de la casa *Promax*, son moduladores *stand alone* (disponen de un panel de control y están instalados en un hardware específico que simplemente necesita ser alimentado). El primero de ellos es un modulador de alta potencia, por lo que encajaría en el segundo caso de los comentados más arriba. Las prestaciones de estos equipos son excelentes, sobretodo en lo que se refiere a la flexibilidad con la que se puede elegir el formato de modulación, así como la frecuencia de emisión de la señal. De este modo, cualquier cambio en el diseño de la red sería fácilmente realizable por estos

equipos, lo que evitaría tener que hacer mayores desembolsos. El modulador *OMB* solamente se ha propuesto para dar una idea real del coste de estos equipos, ya que la oferta de *Promax* es, sin duda, la más barata del mercado. Además, los equipos *Promax* están siendo utilizados con el mismo propósito para el que se proponen en este documento en numerosas televisiones de cable en todo el mundo. Por lo tanto, en este sentido la recomendación a favor de este equipo es evidente.

Los otros dos componentes propuestos son moduladores de baja potencia y que, por lo tanto, necesitan ser amplificados. La solución de la casa *Dektec* es una tarjeta moduladora instalada en un ordenador. Por lo tanto, tiene las mismas limitaciones que se han descrito anteriormente: necesidad de comprar un PC, fiabilidad y estabilidad y espacio ocupado en la cabecera de red. Su única ventaja es ser la solución más barata de todas las propuestas.

La última de las soluciones, basada en productos *Promax*, es la que en principio se recomienda en este estudio, a falta de conocer la opinión de la empresa que se encargue de adaptar la cabecera, por los autores de este documento. Se trata de una estructura para *rack* y pared (*Promax DT-900*) (a partir de ahora nos referiremos a esta estructura como "bastidor") en la cual se pueden integrar diversos elementos. Por un lado, son necesarias una fuente de alimentación y una serie de carátulas ciegas para ocupar los huecos que queden libres en dicha estructura. Por lo demás, se instalarían un modulador de baja potencia de salida (*Promax DT-101*) y un combinador de 8 entradas para TDT (*Promax DT-710*) que será el elemento encargado de amplificar la señal.

Por lo tanto, este equipo nos proporciona una señal a su salida que puede ser mezclada con las señales provenientes de la cabecera e introducida en la red. Sin embargo, el modulador **DT-101** no permite su reconfiguración en frecuencia ni es demasiado flexible a la hora de realizar la modulación. Además, el precio total del conjunto es prácticamente el mismo que el del modulador *stand alone* y la mayoría de la estructura está desaprovechada ya que solamente se instalaría un modulador en ella. Entonces, por qué se propone esta solución? El motivo es que esta estructura permitiría servir como base para otra serie de elementos que serán necesarios a la hora de adaptar la cabecera para el apagón analógico, de tal forma que podría ser utilizada como base por la empresa que realice dicho cometido.

En caso de que dicha empresa prefiera trabajar con componentes de otros fabricantes, el modulador **DT-101** debería ser adquirido por separado, aunque su alimentación y la amplificación de su salida se podrían complicar. Evidentemente, también existe la posibilidad de que sea la empresa que diseñe la ampliación de la cabecera la que tome la decisión en relación al modulador a emplear para el canal local. La única restricción que se dispone en este documento es que dicho modulador disponga de una entrada para tramas de transporte en *DVB-ASI*.

En cualquier caso, y como se ha comentado al iniciar esta descripción, estos aspectos deberían ser puestos en común con la empresa que realice la adaptación de la cabecera.

### 2.3.2 Solución Nº 2: Diferentes soluciones que integran varios componentes en diferentes dispositivos

Esta segunda solución se incluye en una sección diferente a la primera, aunque en realidad la propuesta es muy parecida. La única diferencia entre ambas es que en este caso varios de los componentes individuales que se han descrito se encuentran integrados en un único dispositivo. Además, la solución se divide en varias ofertas diferentes, ya que la forma en que los componentes se integran varía de unas a otras.

El objetivo de esta segunda solución es, por lo tanto, reducir en lo posible el precio de la primera a costa de perder flexibilidad en algunos de los procesos, ya que, por lo general, la integración de componentes disminuye las opciones de reconfiguración del dispositivo. En cualquier caso, son soluciones perfectamente válidas para las especificaciones de este proyecto.

Como se verá, la reducción de precio que finalmente se consigue no es demasiado significativa.

Con el objetivo de simplificar las tablas de presupuestos, para las diferentes opciones, en adelante solo se incluirá el componente recomendado en los apartados de *playout* de vídeo, servidor de aplicaciones, codificador y modulador en los casos en que estos no se encuentren integrados en algún otro dispositivo. En definitiva:

- Ninguna de las propuestas que se muestran a continuación implica la integración del *playout* de vídeo en ningún otro componente. Por lo tanto, todas las consideraciones hechas al respecto del mismo en la sección 2.3.1 son perfectamente válidas. En consecuencia, solo se incluye el *playout* de la casa *Stream Labs*, aunque cualquiera de los otros analizados serviría igualmente.
- El codificador utilizado será siempre el **Sivac 3000 MVE2010-D4A**.
- Lo mismo se aplica para el modulador, en los casos en que este no se integra en ningún otro componente, mostrándose únicamente la opción para bastidor de *Promax*.
- Finalmente, también se muestra únicamente la opción del *MIT Xperts iMUX* como servidor de aplicaciones.

La primera posibilidad, mostrada en la tabla 2ª, consiste en el empleo de moduladores integrados en el propio multiplexor.

Concepto	Precio
<i>Playout</i> de contenidos audiovisuales	
Stream Labs, <b>Wind SDI II</b> [StrP1]	<b>2.100 €</b> <b>+ PC</b>
Codificadores <i>MPEG-2</i>	
Saptec, <b>Sivac 3000 MVE3010-D4A</b> [SapP1]	<b>3.961 €</b>
Servidor de aplicaciones MHP	
MIT Xperts, <b>iMUX Multiplexer</b> [MitP1]	<b>9.251 €</b>
Multiplexadores con modulador integrado	
Abe, <b>DVM 1000/T</b> [AbeP1]	<b>7.250 €<sup>R</sup></b>
AppearTV, <b>MC 3000 Multiplexer+Modulator</b> [AppP1]	<b>11.000 €</b>
Dektec, <b>2xDTA-105 + DTC-700 MuxXpert + DTA-110T</b> [DekP7][DekP5][DekP2]	<b>3.930 €</b> <b>+ PC</b>
Eurotek, <b>EK-PWS/11 + EK-AMX/4+EK-DTX/1</b> [EuP1]	<b>8.200 €</b>
<b>Total (menor precio):</b>	<b>19.242 €</b> <b>+ 2xPC</b>
<b>Total (Solución recomendada<sup>R</sup>):</b>	<b>22.562 €</b> <b>+ PC</b>

**Tabla 2a. Presupuesto Solución N° 2a (precios sin IVA).**

Como se puede observar, empleando una serie de tarjetas *Dektec* será posible reducir sensiblemente los costes derivados de multiplexador y modulador. Sin embargo, la reducción en precio con respecto a la más siguiente opción (*ABE*) no parece recomendar el uso de esta solución, por los problemas derivados de su funcionamiento bajo un ordenador convencional. Además, en ninguno de los casos se consigue una reducción significativa sobre la arquitectura inicial, por lo que no parece que tenga demasiado sentido.

Una segunda opción consiste en integrar el codificador, el multiplexador y el modulador. La tabla 2b detalla el presupuesto asociado a dicha opción.

Concepto	Precio
Playout de contenidos audiovisuales	
Stream Labs, <b>Wind SDI II</b> [StrP1]	<b>2.100 € + PC</b>
Servidor de aplicaciones MHP	
MIT Xperts, <b>iMUX Multiplexer</b> [MitP1]	<b>9.251 €</b>
Codificadores con multiplexador y modulador integrado	
Abe, <b>DME1001-t + DS-HSD +DS-HAI + DS-HUP</b> [AbeP2-3]	<b>9.650 €<sup>R</sup></b>
Eurotek, <b>EK-PWS/11 + EK-AMX/4+EK-DTX/1 + EK-ENS/2</b> [EurP1]	<b>13.200 €</b>
OMB, <b>MOT 5 DVB-T Compact</b> [OmbP1]	<b>7.500 € + Audio Analógico</b>
<b>Total (menor precio):</b>	<b>18.851 € + PC</b>
<b>Total (Solución recomendada<sup>R</sup>):</b>	<b>21.001 € + PC</b>

**Tabla 2b. Presupuesto Solución N° 2b (precios sin IVA).**

Como se puede observar, la casa *OMB*, que por cierto tiene sede en Zaragoza, dispone de un instrumento con un precio excelente y que integra un codificador con un multiplexor y un modulador. De este modo, se puede obtener una reducción cercana a 4.000 euros en relación a la solución N° 1. Sin embargo tiene una limitación muy fuerte que hace que no sea la más recomendable. En concreto, se trata de que el audio a la entrada del instrumento debe ser analógico, a pesar de que el instrumento dispone de una entrada para vídeo SDI. Por lo tanto, será necesario transformar el audio digital que contiene el material audiovisual a analógico, para posteriormente ser digitalizado por este instrumento. Evidentemente esto supondrá una pérdida de calidad que en principio no se debería asumir.

Por lo tanto, la mejor opción en este caso pasa a ser la de utilizar la solución de la casa *ABE* la cual nos lleva a una ligera mejora en el precio final del producto.

Por último, también será posible integrar el codificador en el multiplexor, manteniendo el esquema de modulación tal y como ha sido descrito en la sección 2.3.1. El presupuesto asociado a esta solución se muestra en la tabla 2c.

Concepto	Precio
<i>Playout</i> de contenidos audiovisuales	
Stream Labs, <b>Wind SDI II</b> [StrP1]	<b>2.100 € + PC</b>
Servidor de aplicaciones MHP	
MIT Xperts, <b>iMUX Multiplexer</b> [MitP1]	<b>9.251 €</b>
Codificador con multiplexador integrado	
Abe, <b>EMX1001 + DS-HSD + DS-HAI</b> [AbeP2]	<b>6.150 €</b>
Moduladores COFDM	
Promax, <b>DT-800 + DT-900 + DT-101 + DT-710 + DT-901</b> [ProP1]	<b>3.042 €</b>
<b>Total (Solución recomendada):</b>	<b>20.543 € + PC</b>

**Tabla 2c. Presupuesto Solución Nº 2c (precios sin IVA).**

Esta solución está basada en un instrumento de la casa *ABE*, mediante el cual se consigue una reducción del precio en relación a la solución Nº 1 de aproximadamente 2.000 euros.

En cualquier caso, si se opta por una opción con componentes integrados como las descritas en esta sección, la recomendación por parte de los autores de este informe sería el adoptar la segunda de las presentadas, integrando el codificador, el multiplexor y el propio modulador en un único instrumento. De este modo, el espacio requerido para incluir el canal propio en la cabecera de televisión se reduce considerablemente, y la configuración de todos los instrumentos también se simplifica. La reducción en coste en relación a la solución completa no parece recomendar la pérdida de flexibilidad que se sufrirá, pero el espacio y la simplicidad también son aspectos a tener muy en cuenta.

### **2.3.3 Solución Nº 3: Inyección de contenidos a partir de un servidor de aplicaciones empleando tecnología de *streaming***

En la presente sección se aborda la problemática de la implantación de un canal de televisión digital con capacidad para albergar aplicaciones interactivas desde un punto de vista que, cuando menos, podría considerar innovador.

Por lo general, los servidores de aplicaciones interactivas tienen la única función de, como su propio nombre indica, servir las aplicaciones interactivas. En ese sentido, su único cometido es el de generar una trama de transporte que albergue las aplicaciones MHP a transmitir y que cumpla con el estándar DVB sobre una interfaz ASI. Luego será un multiplexor el que se encargue de incorporar estas aplicaciones a una trama de transporte con los contenidos audiovisuales.



Esto es así ya que, históricamente, el multiplexador es el encargado de juntar varios servicios de televisión ("servicio" es el nombre técnico para lo que los usuarios finales vienen a llamar "canal de televisión") por lo que la multiplexación de los mismos resulta tan compleja que, por lo general, requiere de un instrumento independiente para realizarla.

Por otra parte, en los últimos años se han diseñado servidores de aplicaciones que incorporaban la funcionalidad de incluir un pequeño vídeo dentro de su trama de transporte. Por lo general, este debía ser un pequeño vídeo, previamente codificado en *MPEG-2*, y que se transmitía en bucle.

Sin embargo, los avances en la potencia de los dispositivos informáticos han posibilitado la incorporación de una nueva tecnología a los servidores de aplicaciones MHP. Se trata de la posibilidad de incluir dentro de su trama de transporte un vídeo procedente de un puerto UDP, o lo que es lo mismo, un vídeo transmitido por *streaming*.

El *streaming* de vídeo es una tecnología relativamente nueva y que tradicionalmente se ha venido utilizando orientada a dispositivos informáticos de uso doméstico, como ordenadores personales o receptores de televisión por IP. Dicha tecnología puede ser implementada mediante diversas plataformas (*Windows Media*, *Real Media*, *Quicktime*, *Flash...*), las cuales disponen de software específico para realizar cada una de las tareas que componen la cadena de transmisión, desde los servidores de *streaming* hasta los receptores en el lado del cliente, pasando por herramientas de codificación y compresión.

La gran ventaja de incorporar un flujo de vídeo por streaming en el caso que nos ocupa, es que el control del contenido de dicho flujo de vídeo puede ser realizado empleando herramientas software que, en principio, están orientadas a un entorno web. Además, existen herramientas de libre distribución que podrían acometer esta tarea, con lo que el precio final de la solución se reduce drásticamente, ya que no son necesarios ni el multiplexador, ni el codificador ni la plataforma comercial de *playout* y continuidad.

Como software para gestionar la emisión de contenidos audiovisuales a través de *streaming* se propone el uso herramientas desarrolladas por la plataforma **VideoLAN** del **VLC** en combinación con el **VLM** una utilidad para gestionar la parrilla y escaleta [Vlc1, Vlm1].

*VideoLAN* es una solución de software completa para transmisión de vídeo, desarrollada por estudiantes de *École centrale Paris* y desarrolladores de todo el mundo, dentro de GNU General Public License (GPL). *VideoLAN* está diseñado para transmitir vídeo MPEG en redes con gran capacidad de ancho de banda.

**VLC media player** (también conocido como VideoLAN Client, o **VLC**) es un reproductor multimedia del proyecto *VideoLAN*; es un software libre distribuido bajo la licencia GPL. Soporta muchos códecs de audio y video, así como diferentes tipos de archivos, además de DVD, VCD y varios protocolos streaming. También puede ser utilizado como servidor en *unicast*



o *multicast*, en IPv4 o IPv6, en una red de banda ancha. Utiliza la biblioteca codec *libavcodec* del proyecto *FFmpeg* para manejar los muchos formatos que soporta, y emplea la biblioteca de descifrado DVD *libdvdcss* para poder reproducir los DVD cifrados. Además **VLC** tiene soporte para *Video4Linux*.

En definitiva,

A día de hoy en el mercado existen dos servidores de aplicaciones MHP que incorporan la funcionalidad anteriormente descrita:

- TmBroadcaster de la casa Tmira
- iMUX de la casa MIT Xperts

En ambos casos, esta funcionalidad acaba de ser incluida y solo está disponible en la última versión del software. Además, los propios desarrolladores de estas plataformas confiesan que su desarrollo responde más a entornos de laboratorio que a grandes cadenas de televisión. En definitiva, ninguna cadena de televisión a día de hoy utiliza esta tecnología en sus emisiones. Sin embargo, desde el CEIN y la UPNA estamos convencidos de que su puesta en marcha en el Ayuntamiento de Aranguren sería plenamente satisfactoria.

La tabla 3 detalla el presupuesto necesario para la implementación de esta solución.

Concepto	Precio
<i>Playout</i> de contenidos audiovisuales, vía Streaming	
VLC y VLM	<b>Libre Distribución</b>
UPNA-CEIN iMUX Video Scheduler	<b>Desarrollado por UPNA y CEIN</b>
Servidor de aplicaciones MHP	
MIT Xperts, <b>iMUX Multiplexer</b> [MitP1]	<b>9.251 €<sup>R</sup></b>
Moduladores COFDM	
Promax, <b>DT-800 + DT-900 + DT-101 + DT-710 + DT-901</b> [ProP1]	<b>3.042 €<sup>R</sup></b>
<b>Total (Solución recomendada<sup>R</sup>):</b>	<b>12.293 €</b>

**Tabla 3. Presupuesto Solución N° 3 (precios sin IVA).**

Como se puede observar, el único coste para el proyecto sería el de la adquisición del **iMUX**, ya que el modulador es un gasto inevitable si se

desea introducir la señal en la red. La posibilidad de realizar el mismo proceso mediante **TmBroadcast** de *Tmira* sería mucho más cara y, como se comentó anteriormente, solo tendría sentido en caso de que se contrate a esta empresa para realizar el desarrollo de las aplicaciones interactivas.

Una funcionalidad añadida de esta solución consiste en la posibilidad de integrar un segundo servicio de vídeo dentro del mismo *multiplex* (o canal físico). **IMUX** permite la inclusión de varios canales en la trama de transporte a su salida. Por lo tanto, sería posible la generación de dos canales, cada uno de los cuales tendría la siguiente función:

- **Canal Aranguren 1:** Distribución del contenido audiovisual propio del ayuntamiento de Aranguren, realizando una escaleta o parrilla de dicho contenido de tal forma que las emisiones puedan ser programadas en el tiempo. La implementación de la escaleta se realizaría por tecnología *streaming* y se controlaría con el software **VLC**.
- **Canal Aranguren 2:** Distribución de un segundo canal de televisión propio del Ayuntamiento del Valle de Aranguren, en el que únicamente se transmitiría, en bucle, un vídeo describiendo los servicios ofrecidos en el punto de información municipal (estos vídeos están siendo generados a día de hoy por el personal del Valle de Aranguren).

Las aplicaciones MHP podrían ser incluidas de tal forma que fueran accesibles desde ambos canales.

Un posible inconveniente de la presente solución estriba en que la escaleta que se podría realizar empleando el software **VLC** no tendría la misma funcionalidad que las escaletas comerciales que se han descrito anteriormente (*VSN, VectorBox, TeleInfo...*). Sin embargo, su funcionalidad es más que suficiente para los propósitos de este proyecto, al menos en sus especificaciones actuales.

En cualquier caso, si en un futuro se quisiera instalar un control más avanzado y un sistema de producción audiovisual profesional, la solución propuesta sería perfectamente escalable por cualquiera de los dos medios que se describen a continuación:

- Compra de un software profesional de *playout* con capacidad de emitir su contenido por *streaming*. La mayoría de las plataformas de continuidad, a día de hoy, no incorporan esta posibilidad. Sin embargo existen muchas plataformas destinadas a la transmisión de vídeo por IP que si lo hacen. En particular, la empresa *Cinegy* estaría muy interesada en introducir sus productos en el Valle de Aranguren, y ha realizado una oferta excelente por sus productos. Para más detalles sobre esta posibilidad, consúltase el apartado 2.5.5 donde se describen los productos de esta empresa.
- Adquisición de el resto de componentes que componían cualquiera de las soluciones descritas en los apartados 2.3.1 y 2.3.2, a las que habría que restar el gasto derivado del **IMUX** y del modulador, los cuales ya habrían sido adquiridos en esta primera fase.

## Otra posibilidad...

Por último, comentar que la UPNA y el CEIN, a lo largo de este verano, han estado trabajando en el desarrollo de un software de *playout* y continuidad propio y que ha sido bautizado como **CEIN-UPNA iMux Video Scheduler**\*. Se trata de una solución software que permite controlar, a partir de la API (*Application Programming Interface*) del **iMux**, esta última plataforma e incorpora las siguientes funcionalidades:

- Permite hacer una parrilla o escaleta a partir de vídeo almacenados en la base de datos de **iMux** controlando, además, la subida de vídeos a la dicha base de datos desde cualquier ordenador con una conexión a internet.
- Permite el control de las aplicaciones MHP a transmitir a través del **iMux**, controlando todos los parámetros establecidos por el mismo, incluyendo la hora de inicio y final, el tipo de aplicación, los permisos de dicha aplicación en función del servicio, su incorporación a un servicio determinado... etc.
- Permite la generación y control de *stream events* (eventos que se insertan en la trama de transporte y permiten la sincronización temporal de las aplicaciones MHP con los contenidos audiovisuales).
- Permite la inserción automática de un vídeo en bucle en los momentos en que no se haya programado ningún vídeo para su emisión (esta funcionalidad se podría aprovechar en el Valle de Aranguren, para insertar el vídeo con el punto de información municipal en los momentos en que no se esté emitiendo ningún otro contenido).

Por lo tanto, se trata de una herramienta software que permite el control de la emisión de contenido audiovisual empleando el **iMux** y por otros medios diferentes a la tecnología de *streaming*. Se incluye en este documento porque constituye una opción perfectamente válida para los objetivos de este proyecto. En caso de que el Ayuntamiento de Aranguren se decantara por la implantación de la solución descrita en este apartado, se procedería a instalar ambas opciones (**VLC** y **CEIN-UPNA iMux Video Scheduler**) de tal forma que sea el Ayuntamiento el que se decante por una u otra.

### 2.3.4 Solución Nº 4: *playout* básico sin aplicaciones interactivas ni gestión avanzada de contenido

Todas las soluciones propuestas hasta el momento incluían entre sus prestaciones la posibilidad de insertar aplicaciones interactivas MHP. Sin embargo, también existen soluciones más modestas y que prescindirían de esa funcionalidad.

---

\* Este software ha sido desarrollado por la estudiante de Ingeniería de Telecomunicación Elena Gadea Aransay, bajo la supervisión de UPNA y CEIN.

Para ello, el esquema de la figura 2 se simplifica hasta el punto de quedar reducido al que se presentó en la figura 3 y al que hay que añadir un modulador. Por lo tanto, se componen de plataforma de *playout* de vídeo, codificador *MPEG-2* en *DVB-ASI* y modulador COFDM. La tabla 4a presenta el presupuesto que se obtiene al eliminar el resto de componentes y quedar con los tres anteriores.

Concepto	Precio
<i>Playout</i> de contenidos audiovisuales	
Stream Labs, <b>Wind SDI II</b> [StrP1]	<b>2.100 €</b> <b>+ PC</b>
Codificadores <i>MPEG-2</i>	
Saptec, <b>Sivac 3000 MVE3010-D4A</b> [SapP1]	<b>3.961 €</b>
Modulador COFDM	
Promax, <b>DT-800 + DT-900 + DT-101 + DT-710 + DT-901</b> [ProP1]	<b>3.042 €</b>
<b>Total:</b>	<b>9.103 €</b> <b>+ PC</b>

**Tabla 4a. Presupuesto Solución Nº 4a (precios sin IVA).**

Como se puede apreciar, la reducción de precio no es demasiado significativa en relación a la oferta presentada en la solución Nº 4. Además, en este caso se hace necesaria la compra de un ordenador para la instalación de la tarjeta de vídeo, por lo que el precio es prácticamente el mismo que en el caso anterior.

En cualquier caso, esto constituye una solución “clásica” para la emisión de un canal propio y, como tal, merece ser tenida en cuenta.

Una segunda opción viene de emplear un componente de la casa *Icareus* como es el servidor de vídeo **VS100 Video Server**. Este es un elemento que viene instalado en su propia unidad de *rack* y que, en principio, está pensado para servir el vídeo hacia el servidor de aplicaciones MHP de la misma casa (**PC100 Compact**). Es un caso bastante particular entre los servidores de vídeo, ya que en lugar de tener una salida SDI dispone directamente de una salida *DVB-ASI*. Por lo tanto, los datos digitales a su salida constituyen una trama de transporte apta para la transmisión y recepción en una red de televisión. De hecho, el hardware de la unidad de *rack* incluye una tarjeta *Dektec DTA-105* como las que se han recomendado en otros puntos de esta memoria. En definitiva, la solución no requiere de un codificador por lo que resulta muy asequible. Además, tampoco es necesaria la adquisición de un ordenador como ocurría en el caso anterior. La tabla 4b presenta el presupuesto asociado a esta solución.

Concepto	Precio
<i>Playout</i> de contenidos	
Icareus, <b>VS100 Video Server</b> [IcaP1]	<b>4.760 €</b>
Modulador COFDM	
Promax, <b>DT-800 + DT-900 + DT-101 + DT-710 + DT-901</b> [ProP1]	<b>3.042 €</b>
<b>Total:</b>	<b>7.802 €</b>

**Tabla 4b. Presupuesto Solución N° 4a (precios sin IVA).**

Otra de las ventajas de esta solución es que si en un futuro se pretende introducir aplicaciones MHP será plenamente compatible con el servidor de aplicaciones MHP que *Icareus* tenga en el mercado en ese momento.

Entre sus desventajas hay que resaltar que el software de continuidad que incorpora es relativamente rudimentario, no estando a la altura ni siquiera del **TeleInfo** de *Stream Labs* aunque, al menos, es capaz de introducir un vídeo en los momentos en que no exista nada programado, lo que garantiza que la emisión no cesa en ningún momento.

### 2.3.5 Solución N° 5: digitalización del canal analógico que se transmite en la actualidad

La solución trivial para la digitalización del canal local analógico que se transmite en la actualidad, consiste en digitalizar la señal analógica que se está transmitiendo y su modulación en formato COFDM para su inserción en la red de cable del valle. En este caso solo serían necesarios dos componentes: un codificador *MPEG-2* con entrada analógica y salida *DVB-ASI* y un modulador COFDM con entrada *DVB-ASI* y salida en RF. En la tabla 5a se muestra el presupuesto necesario para llevar a cabo esta solución empleando un modulador de la casa *Promax*.

Concepto	Precio
Promax, <b>DT-504</b> [ProP1]	<b>3.318 €</b>
<b>Total:</b>	<b>3.318 €</b>

**Tabla 5a. Presupuesto Solución N° 5a (precios sin IVA).**

Evidentemente, se trata de una solución realmente de compromiso, aunque muy barata, mediante la cual el ayuntamiento de Aranguren mantendría su infraestructura actual y su modo de funcionamiento, en relación al punto de información municipal, sin más que añadir un modulador a la entrada de la red.

Una segunda solución muy sencilla para disponer de un canal propio de televisión digital consiste en transmitir en bucle un vídeo almacenado en un PC al que se incorpora una tarjeta moduladora y un sencillo software de

*layout*. En la tabla 5b se muestra el presupuesto asociado a esta segunda solución.

Concepto	Precio
Dektec, <b>DTA-110T</b> [DekP2]  Software: <b>Dektec DTC-300 StreamExpress</b>	<b>995 € + PC</b>
<b>Total:</b>	<b>995 € + PC</b>

**Tabla 5b. Presupuesto Solución Nº 5b (precios sin IVA).**

La diferencia con la anterior es que en este caso lo que se transmite es un vídeo digital que ha sido editado por medios digitales y en cuya cadena de transmisión no se produce ninguna transición a analógico. Por lo tanto, la calidad de la señal emitida será mucho mayor. Además, esta solución viene a dar cabida a la posibilidad de transmitir en bucle, sin más pretensiones, los vídeos que actualmente se están preparando en el Ayuntamiento para sustituir al anterior punto de información municipal.

El software y el hardware necesario para poner en marcha esta solución tienen unas funcionalidades muy limitadas. En particular, se trata de una tarjeta moduladora de baja potencia y sin ninguna entrada digital. El software **DTC-Stream Express** [Dek8] permite la transmisión en bucle, empleando la tarjeta moduladora **DTA-110T**, de una trama de transporte (previamente generada por otro software como el **Mainconcept Reference** [Mai1] o el **VLC** [Vlc1]). Por lo tanto, no es posible realizar una escaleta ni programar la emisión de contenidos.

En cualquier caso, sería posible el cambio del vídeo en emisión, pero para ello es necesario que un operario realice dicha operación en modo manual, lo que resulta poco eficiente y poco fiable, a lo que hay que añadir el hecho de que mientras dura la operación (unos pocos segundos, en el mejor de los casos) la emisión se cortaría.

Lo mismo puede realizarse utilizando un modulador en bastidor como el que se ha venido recomendando hasta ahora. La funcionalidad de esta solución es exactamente la misma a la anterior, con la ventaja de que en el bastidor podrían instalarse los transmoduladores DVB-S a DVB-T a los que se ha hecho mención en la sección 2.3.1 [Pro9]. El presupuesto para esta solución se detalla en la tabla 5c. Como se puede observar el modulador **DTA-110T** ha sido sustituido por una tarjeta **DTA-105** [Dek7] con salida ASI que alimentaría al modulador *Promax*.

Concepto	Precio
Playout de contenidos	
Dektec, DTA-105 [DekP7]	<b>845 €</b>
Software: Dektec DTC-300 StreamExpress	<b>+ PC</b>
Modulador COFDM	
Promax, <b>DT-800 + DT-900 + DT-101 + DT-710 + DT-901</b> [ProP1]	<b>3.042 €</b>
<b>Total:</b>	<b>3.882 €</b>

**Tabla 5c. Presupuesto Solución N° 5c (precios sin IVA).**

### 2.3.6 Solución N° 6: Contratación de una empresa externa que proporcione todos los servicios y se encargue del mantenimiento

A pesar de que en este informe solamente se pretende tratar el tema de la infraestructura necesaria para la implantación del canal local en el Valle de Aranguren, abstrayéndolo totalmente de las aplicaciones interactivas o de los servicios que en un futuro podrían integrarse en el mismo o de cómo implementar dichos servicios, en este apartado se realiza un pequeño inciso en esa línea, para recomendar una solución integral.

En este sentido, existen varias empresas especializadas en la implantación de cadenas de televisión locales o incluso redes de cable en hoteles o urbanizaciones. Estas empresas ofrecen un servicio global, que va desde la instalación de toda la infraestructura necesaria, hasta la programación de los contenidos a transmitir sobre la misma. Por supuesto, también ofrecen servicios de mantenimiento y actualización de las instalaciones o incluso formación para sus clientes en lo relativo a la programación de aplicaciones.

Son servicios relativamente caros y poco flexibles, ya que una vez contratado el producto el cliente no tiene ningún control ni conocimiento real sobre su infraestructura. Además, las aplicaciones interactivas generadas no suelen ser demasiado atractivas para el usuario final; no en vano se trata de aplicaciones tipo "plantilla" y que pueden ser exportadas con facilidad de un cliente a otro.

En cualquier caso, si el Ayuntamiento del Valle de Aranguren optara por esta solución, la recomendación es que contacte con cualquiera de estas dos empresas:

- Activa Multimedia
- Tmira

Más información sobre su oferta puede encontrarse en el apartado 2.5 de este informe o en las propias ofertas realizadas por las empresas y que se encuentran a disposición del Ayuntamiento en el CD adjunto. Evidentemente, el siguiente paso sería realizar una visita a alguna de las



cadena de televisión que hacen uso de sus servicios, con el fin de tener una idea real de en qué consiste su oferta.

Finalmente, y solo por dar una idea del coste de estas soluciones, en la tabla 6 se muestran los presupuestos presentados por las mismas.

Concepto	Precio
Activa Multimedia [ActP1]	<b>135.280 €</b>
Tmira [TmiP1]	<b>50.000 € + Desarrollo aplicaciones</b>
<b>Total:</b>	<b>A negociar con la empresa</b>

**Tabla 6. Presupuesto Solución Nº 6 (precios sin IVA) (precios sin IVA).**

### **2.3.7 Solución Nº 7: Inyección de contenidos con la ayuda de programas de software libre empleando tecnología de streaming**

Como alternativa a la propuesta de la apartado 2.3.3, en la que se intenta aprovechar la capacidad que poseen ciertos servidores de aplicaciones interactivas de recibir por *streaming* los contenidos, existe la posibilidad de usar programas de software libre para adaptar los contenidos al estándar DVB y multiplexarlos con las aplicaciones interactivas.

Para ese propósito existe *Opencaster*. *Opencaster* es un conjunto de herramientas desarrolladas por la compañía italiana *Avalpa* que surgen como culminación del proyecto *JustDVB-IT* difundido bajo licencia GNU *General Public License* (GPL). Estas herramientas están enfocadas a generar, procesar, emitir y difundir contenidos encapsulados en *MPEG-2 Transport stream (MPEG-2 TS)*, que es el estándar usado por *DVB-T*, en un sistema *Linux*

*Opencaster* es capaz al igual que ciertos servidores de aplicaciones interactivas, de recibir por *streaming* los contenidos para su posterior procesamiento.

Así pues, es interesante el uso de herramientas de libre distribución como el *VLC* con su aplicación *VLM*, para gestionar la escaleta, en combinación con las herramientas de *Opencaster* para la adaptación de los contenidos y la multiplexación con las aplicaciones interactivas, para desarrollar el mismo esquema que en el apartado 2.3.3.

Por un lado el uso de *Opencaster* permitiría prescindir del multiplexor con lo que el coste descendería de forma considerable. Por otro lado la mayor desventaja a la hora de usar *Opencaster* es la falta de un interfaz gráfico de usuario (*GUI Graphic User Interface*) y el hecho de ser unas herramientas desarrolladas para un sistema *Linux*. El sistema operativo *Linux* es cada vez más amigable y tiende a ser cada vez más visual, pero su uso no está muy

extendido ante la gran mayoría de usuarios y éstos no están acostumbrados al manejo del mismo.

De esta forma, utilizar software libre para la gestión de la escaleta y la puesta en emisión reduce de forma considerable el coste del sistema, además, sendos programas podrían ser instalados en un mismo PC. Por otro lado a los inconvenientes del uso de VLC explicados en el apartado 2.3.3 se unen los inconvenientes de uso de *Opencaster*.

Concepto	Precio
Playout de contenidos audiovisuales, vía Streaming	
VLC y VLM	<b>Libre Distribución + PC</b>
Multiplexador MHP	
Opencaster	<b>Libre Distribución + PC</b>
Moduladores COFDM	
Dektec, <b>DTA-110T</b> [DekP2]	<b>995 € + PC</b>
Promax, <b>DT-800 + DT-900 + DT-101 + DT-710 + DT-901</b> [ProP1]	<b>3.042 €</b>
<b>Total (menor precio):</b>	<b>995 € + PC</b>
<b>Total (Solución recomendada):</b>	<b>3.042 €</b>

Una funcionalidad añadida de esta solución consiste en la posibilidad de integrar un segundo servicio de vídeo dentro del mismo *multiplex* (o canal físico) como en la solución Nº 3. *Opencaster* permite la inclusión de varios canales en la trama de transporte a su salida. Por lo tanto, sería posible la generación de los dos canales sugeridos en el apartado 2.3.3.

La posible escalabilidad del proyecto queda también asegurada ya que si en un futuro se quisiera instalar un control más avanzado y un sistema de producción audiovisual profesional, la solución propuesta sería perfectamente escalable los medios que se describen a continuación:

- Compra de un software profesional de *playout* con capacidad de emitir su contenido por *streaming*. La mayoría de las plataformas de continuidad, a día de hoy, no incorporan esta posibilidad. Sin embargo existen muchas plataformas destinadas a la transmisión de vídeo por IP que si lo hacen. En particular, la empresa *Cinegy* estaría muy interesada en introducir sus productos en el Valle de Aranguren, y ha realizado una oferta excelente por sus productos. Para más

detalles sobre esta posibilidad, consúltase el apartado 2.5.5 donde se describen los productos de esta empresa.

- Adquisición de el resto de componentes que componían cualquiera de las soluciones descritas en los apartados 2.3.1 y 2.3.2, a las que habría que restar el gasto derivado del modulador, los cuales ya habrían sido adquiridos en esta primera fase en caso de optar por la compra del modulador.
- La opción de elegir la Tarjeta moduladora *PCI* para el PC no es escalable ya que al instalar nuevos equipos la Tarjeta quedará aislada dentro del sistema.

Esta última solución ha sido estudiada en el laboratorio de Televisión gracias a su bajo coste y a que se disponía del hardware necesario para su funcionamiento y posterior prueba. En los capítulos 3, 4, 5 y 6 se expondrán todos los aspectos a tener en cuenta a la hora de implementar este sistema de forma práctica.

## Receptores

En el apartado 1.3 se introdujeron las características de un receptor de televisión digital ideal preparado para afrontar las diferentes transiciones tecnológicas que tendrán lugar en los próximos meses. De este modo, como parte del trabajo realizado en este proyecto se ha realizado una búsqueda exhaustiva de los receptores, buscando que cumplieran las siguientes características:

- Canal de retorno con preferiblemente *Ethernet* que permita dotar a la aplicación interactiva de un canal bidireccional de comunicación para la personalización de los servicios interactivos. En muchos casos el canal de retorno disponible será mediante *modem* telefónico V90, lo que ralentiza mucho la comunicación con las localizaciones remotas, además de bloquear la línea telefónica del usuario.
- Capacidad para reproducir aplicaciones MHP.
- Receptor con soporte la recepción de contenidos en alta definición para la recepción futura de canales de alta definición.
- Receptor que disponga de un sistema de acceso condicional actualizable universal, de tal forma que el usuario pueda elegir el la plataforma de acceso condicional que desea incorporar en el mismo. De este modo, se evita la compra de un nuevo receptor en el momento en que las cadenas decidan modificar el tipo de acceso condicional. Aunque esta funcionalidad ha sido definida por Ministerio de Industria, Turismo y Comercio [Min1], es altamente improbable que lleguen a fabricarse receptores tan versátiles. En cualquier caso, lo ideal será la adquisición de un receptor cuyas prestaciones se acerquen lo máximo posible a esta "utopía".
- Opcionalmente, podría ser interesante que el receptor estuviera dotado de lector de tarjeta *Smartcard* (estándar ISO 7816-4) que permitan integrar tarjetas con chip para su utilización dentro de la aplicación interactiva, aunque esta funcionalidad no ha sido contemplada en las aplicaciones que se proponen en el segundo documento adjunto [Ces2].

En lo relativo a la interactividad, es importante puntualizar que existen varias versiones diferentes del estándar de codificación de aplicaciones interactivas MHP. A continuación se muestra una tabla con las diferentes versiones de MHP y sus características principales.

Versión MHP	Tipo de aplicaciones	Soporte	Extras
<b>1.0.x</b>	Aplicaciones ligadas al canal de televisión	Datos vía IP	
<b>1.1.x</b>	Almacenamiento de aplicaciones en el receptor (STB)	<ul style="list-style-type: none"> <li>Datos vía IP</li> <li>Aplicaciones vía IP</li> </ul>	<ul style="list-style-type: none"> <li>Soporte HD</li> <li>Smartcard</li> <li>Navegador básico HTML</li> </ul>
<b>1.2.x</b>	<ul style="list-style-type: none"> <li>Aplicaciones no asociadas al canal de televisión</li> <li>Aplicación con privilegios de gestión</li> </ul>	<ul style="list-style-type: none"> <li>Datos vía IP</li> <li>Aplicaciones vía IP</li> <li>Vídeo vía IP</li> </ul>	<ul style="list-style-type: none"> <li>DVB-IPTV</li> <li>Vídeo On Demand</li> <li>GEM-IPTV</li> </ul>

A día de hoy no existen receptores que implementen la versión 1.2 del estándar MHP. En realidad, la mayoría de los que existen en el mercado cumplen con MHP 1.0.2, lo que limita en gran medida su funcionalidad, aunque son perfectamente válidos para la reproducción de las aplicaciones MHP que se proponen en el segundo documento adjunto [Ces2]. Sin embargo, lo ideal sería disponer de receptores MHP de última generación (la versión comercial más avanzada es la MHP 1.1.3), pero como se podrá observar a continuación estos tampoco estarán disponibles en el mercado español hasta mediados del año que viene. Por lo tanto, se recomienda la adquisición de receptores que dispongan de la pila *middleware* MHP en su versión 1.1.2 o posterior, y a ser posible que sean actualizables por el aire (OTA – *Over the air*). El OTA es una funcionalidad que incorporan algunos receptores y que le permite actualizar su *middleware* sin necesidad de la presencia de un técnico en la vivienda del usuario. De este modo, el receptor actualiza sus funcionalidades directamente a partir del propio canal de televisión. Debido de su amplia experiencia y conocimiento del mercado, lo óptimo sería la adquisición de dichos decodificadores con *middleware* *Osmosys* [Osm1] o *Alticast* [Alt1].

Finalmente, se muestran los diferentes receptores disponibles en el mercado, ordenados por empresas, que podrían servir, en mayor o menor medida, a los propósitos de este proyecto.

### 2.3.8 ADB-Osmosys

- Web: [www.adbglobal.com](http://www.adbglobal.com), [www.osmosys.tv](http://www.osmosys.tv)
- Contacto:
  - Albert Cañigüeral (technical marketing, [a.canigueral@osmosys.tv](mailto:a.canigueral@osmosys.tv), 649842088)

- Daniel Milla (VP Sales manager Iberian Region, d.milla@adbgloal.com).

El grupo *ADB* al que pertenecen *ADB* y *Osmosys* fue fundado en 1995, *ADB* se centró inicialmente en el desarrollo y comercialización de software para los procesadores más avanzados de televisión digital. En 1997 el negocio se amplió en el diseño y fabricación de equipos de televisión digital.

Se trata de una empresa puntera en el sector que ha ido creciendo continuamente estableciendo oficinas en Australia, Italia, Polonia, España, Suiza, Taiwán, Tailandia, Ucrania, el Reino Unido y los Estados Unidos.

La fortaleza del grupo se basa en la continua búsqueda de la innovación, teniendo así más del 60% de su personal dedicado a la investigación y desarrollo.

El grupo opera en tres principales segmentos de negocio:

- Equipo de TV Digital: diseño, fabricación, comercialización y venta de equipos locales de los consumidores, incluidos los *set-top boxes* y los módulos de televisión integrada. Este segmento opera fundamentalmente en el marco de *ADB* y del *Grupo I-can*.
- Software y servicios: los productos de software, las licencias de *middleware*, herramientas de desarrollo, y servicios de verificación, integración y otros servicios. Este segmento opera fundamentalmente en el marco *Osmosys* y *Vidiom*.
- Nuevas iniciativas: aplicaciones interactivas, nuevos medios, contenidos y servicios. Este segmento opera fundamentalmente en el marco del *Grupo Simple* y *tele.DOM*.

### **Soluciones aportadas por la empresa al esquema global:**

- STB receiver

En la reunión que se produjo en el IBC2009 [Ibc1], Daniel Milla muestra mucho interés en el proyecto, siempre que se plantee como un tema de desarrollo, prueba piloto o en algún esquema de ese tipo. Le interesa entrar en el mercado español y hacer un poco de publicidad con el tema.

En relación a los receptores, comenta que las especificaciones que se buscan para el Valle de Aranguren son imposibles de cumplir, de aquí a diciembre, para un receptor de cable en castellano. Comentan la posibilidad de traer receptores destinados al mercado italiano o noruego, aunque los menús estarían en italiano o noruego e inglés [Adb1 ] (desde luego, esto le daría un gran aire "experimental" al tema...).

A continuación presentamos en una tabla los receptores de la casa *ADB* [Adb2-4]:

Modelo	Versión MHP	Acceso Condicional	HD	Canal de retorno	Middleware	Precio
<i>ADB-3800</i>	1.1	A elegir	si	10/100BT port ethernet	Osmosys	≈1000
<i>ADB-5100TX</i>	1.1	A elegir	No	10/100BT port ethernet	Osmosys	≈1000
<i>ADB-5810TX</i>	1.1	A elegir	si	10/100BT port ethernet	Osmosys	≈1000
<i>DTR 8740</i>	1.1	A elegir	si	10/100BT port ethernet	Osmosys	150

### 2.3.9 Digiquest

- Web:
  - [www.digiquest.it](http://www.digiquest.it)
  - [www.cardmania.it](http://www.cardmania.it)
- Contacto: info@digiquest.it

*Digiquest* es una empresa con sede en California (Estados Unidos). Tienen como distribuidor en Italia a *Cardamia*. Es una empresa especializada en receptores de satélite y terrestre. Tienen un gran despliegue en Italia donde han sacado numerosos receptores compatibles con MHP.

A continuación presentamos en una tabla los receptores de la casa *Digiquest* [Dig1][Dig2][Dig3]:

Modelo	Versión MHP	Acceso Condicional	HD	Canal de retorno	Middleware	Precio
<i>MHP 3007</i>	1.0.x	Irdeto, Nagravision y conax	No	Modem V.90	Osmosys	69
<i>MHP 3008</i>	1.0.x	Irdeto, Nagravision y conax	No	Modem V.90	Osmosys	69
<i>MHP 2007</i>	1.0.x	Irdeto, Nagravision y conax	No	Modem V.90	Osmosys	

### 2.3.10 Engel

- Web: [www.engel.es](http://www.engel.es)
- Contacto: Jon Zagarzazu (JSagarzazu@engel.es)

*Engel* es una empresa francesa con sede en Perpignan, pero tiene distribuidores repartidos por toda España. Dispone de una larga lista de productos en el sector de las telecomunicaciones como receptores, moduladores, amplificadores conversores RF, etc... todos ellos dedicados al sector audiovisual.



A continuación presentamos en una tabla los receptores de la casa:

Modelo	Versión MHP	Acceso condicional	HD	Canal de retorno	Middleware	Precio
6000i	1.1.2	no	no	modem V.90	Osmosys	129

### 2.3.11 Humax

- Web: [www.humaxdigital.com](http://www.humaxdigital.com)
- Contacto: sales-info@humax-digital.co.uk

*Humax* es una empresa con sede en Corea pero con oficinas en Dubai, Alemania, India, Italia, Japón, Tailandia, Australia, Hong Kong, Suecia, España, el Reino Unido y los Estados Unidos.

Es una empresa con una larga trayectoria, lo que le ha llevado a ser uno de los mayores distribuidores de receptores del mundo y a día de hoy distribuye a más de 90 países.

Ha estado muy inmerso en la conversión a televisión digital lo que le ha llevado a hacer ensayos en Reino Unido en 2006 y estar en el lanzamiento la *premiere* de la alta definición.

A continuación presentamos en una tabla los receptores de la casa[Hum1][Hum2]:

Modelo	Versión MHP	Acceso Condicional	HD	Canal de retorno	Middleware	Precio
<i>Digifox</i>	1.0.3	No	No	Modem V.90	Alticast	*
<i>DTT-3500</i>	1.0.2	Irdeto, Nagravision y conax	No	Modem V.90 / V.34	Alticast	*
<i>DTT-3600</i>	1.0.3	Irdeto, Nagravision y conax	No	Modem V.90	Alticast	*
<i>Tivubox</i>	1.0.3	Irdeto, Nagravision y conax	No	Modem V.90	Alticast	*
<i>DTT-Nano</i>	1.0.3	No	No	Modem V.90	Alticast	*

\*El precio debe ser negociado con distribuidor según el numero de unidades

### 2.3.12 I-cant

- Web: <http://www.i-can.tv/it/index.htm>
- Contacto: Vía formulario web

Es una empresa que está dentro el *ADB*, lo cual le hace pertenecer a uno de los grupos más importantes del sector y por lo tanto tiene distribuidores por todo el mundo. Se podría decir es el fabricante económico de *ADB*. En sus productos siempre utilizan estándares abiertos lo que les da gran versatilidad.

A continuación presentamos en una tabla los receptores de la casa[Icon1][Icon2]:

Modelo	Versión MHP	Acceso condicional	HD	Canal de retorno	Middleware	Precio
2000T	1.0.x	Nagra, Irdeto o Conax	No	Modem V92	Osmosys	*
3810T	1.1.2	Nagra, Irdeto o Conax	Si	10/100BT port ethernet	Osmosys	150

\*Consultar precio con el distribuidor, en este caso *el corte ingles* para pedidos grandes

### 2.3.13 Invest

- Web: [www.elcorteingles.es/tiendas\\_e/cda/investronica/scd/0,5631,PD16055!INVESTRON,00.html](http://www.elcorteingles.es/tiendas_e/cda/investronica/scd/0,5631,PD16055!INVESTRON,00.html)
- Contacto:

*Inves* es la marca española de equipos informáticos fabricados por *Investrónica* y fundada en 1986. Esta ha ido creciendo hasta convertirse en uno de los principales actores del mercado TI español, con más de un millón de dispositivos vendidos. La planta de fabricación de la organización está ubicada en la localidad de Tres Cantos y su Centro Logístico en Getafe, ambas en Madrid. El distribuidor del *Invest* es el Corte Inglés

A continuación presentamos en una tabla los receptores de la casa[Inv1][Inv2][Inv3]:

Modelo	Versión MHP	Acceso condicional	HD	Canal de retorno	Desmodulación	Precio
2200M	1.0.3	No	No	modemV.90	Osmosys	115
2300M	1.0.3	No	No	modemV.90	Osmosys	79
3250E	1.1.2	No	No	10/100BT port ethernet	Osmosys	120

### 2.3.14 Illusion

- Web: <http://www.tecatel.com/>

A continuación se presentan los receptores de la casa illusion:

Modelo	Versión MHP	Acceso Condicional	HD	Canal de retorno	Middleware	Precio
RCP-ILLMHP100T	1.0.2	No	No	modemV.90	No especificado	130
RCP-ILLMHP200T	1.0.2	No	No	modemV.90	No especificado	102.5

### 2.3.15 Strong

- Web: [www.strongsat.com](http://www.strongsat.com)
- Contacto: victor.planas@afex.es , teléfono: +34 629771337

*Strong* es una compañía dentro del grupo *Strong & CO* establecido en Yokohama (Japón) desde 1868. En la actualidad es uno de los principales proveedores de Europa de receptores de televisión. Su distribuidor en España es la *fnac* lo que le asegura tener puntos de distribución por toda ella.

A continuación se presentan los receptores de la casa Strong[Stro1][Stro2][Stro3]:

Modelo	Versión MHP	Acceso condicional	HD	Canal de retorno	Middleware	Precio
STR-5501	1.1.2	No	No	modemV.90	Osmosys	*
STR-5502	1.1.2	No	No	modemV.90	Osmosys	*
STR-5510	1.1.2	No	No	puerto ethernet (RJ45)	Osmosys	69

A negociar con la distribuidora según el numero de unidades.

## 2.4 Relación de empresas y descripción de las soluciones aportadas por las mismas a cada una de las partes del esquema global

### 2.4.1 ABE

- Web: [www.abe.it](http://www.abe.it)
- Contacto:
  - Matteo Ripamonti ([matteo.ripamonti@abe.it](mailto:matteo.ripamonti@abe.it))
  - Urbano Pagnoncelli ([urbano.pagnoncelli@abe.it](mailto:urbano.pagnoncelli@abe.it))

*ABE Elettronica* es una compañía italiana fundada en 1979 con sede en Treviglio aunque posteriormente se trasladó a Caravaggio entre 1980 y 1981. Sus siglas son de *Advanced Broadcasting Electronics*, y fueron elegidas para reflejar perfectamente la vocación de la compañía por la innovación, haciendo importantes inversiones en I+D.

ABE fue desarrollada durante el período de la expansión del *broadcasting* en Italia en los años 80 y se encuentra actualmente entre los más reconocidos y apreciados productores de transmisores televisivos, sistemas de comunicación de microondas y codificadores MPEG.

Durante sus 30 años, ABE ha ido desarrollando y produciendo una gama completa de equipos de televisión, analógicos y digitales: codificadores *MPEG-2* y multiplexores, transmisores de televisión y retransmisores, enlaces de microonda de hasta 24 GHz, *uplinks* al satélite, antenas y sistemas radiantes [Abe1].

Hoy en día la empresa se encuentra operando en más de 80 países del mundo.

#### Soluciones aportadas por la empresa al esquema global:

- Multiplexer
- COFDM Modulator
- COFDM Modulator
- Otros (codificador)

**Serie MUX1000** de multiplexores [Abe4]. Tiene la posibilidad de insertar hasta 5 tarjetas a la entrada del multiplexor, siendo el software de control el mismo en todos los casos. La quinta "pestaña" de la interfaz de usuario de este software es la que permite controlar los parámetros de multiplexación de una trama MHP.

Este multiplexor es capaz de guardar configuraciones para cargarlas posteriormente, pero siempre trabajando en modo manual: no es posible programar el cambio de configuración (*scheduling*).

En la configuración básica que se pretende implantar en el Valle de Aranguren, solo serían necesarias dos entradas y una salida, por lo que con el **MUX1002** es suficiente.

Modulador COFDM **DVM 1000/T** [Abe5]. Modulador que cuenta con una señal de entrada ASI, aunque también tiene como opción otras interfaces de entrada: GBE (10/100/1000 Ethernet - Vídeo sobre IP), receptor de satélite (DVB-S/S2) o receptor Terrestre (DVB-T).

La salida puede ser en Frecuencia intermedia (IF, de acuerdo al estándar, a 36 o 44 o 70MHz) o en VHF, UHF o banda L.

Dispone de SNMP AGENT para mandar alertas del estado del modulador además de un software para el control remoto del mismo.

Codificador *MPEG-2* **EMX 1001** [Abe6]. Tiene la posibilidad de insertar de 1 hasta 4 entradas de video *MPEG-2* en tiempo real que multiplexa en un único *stream* de datos

El sistema codificador puede codificar hasta una resolución de 720p horizontal y 576p en vertical y las resoluciones soportadas son: Full D1, 3/4 D1, 2/3 D1, 1/2 D1, SIF, QSIF.

Codificador *MPEG-2* con modulador COFDM integrado **DME 1001/T** [Abe4]. Este *rack* incluye el codificador **EMX 1001** más el multiplexor **MUX1000** más el modulador **DVM 1000/T** todo en uno. Las características son iguales que sus componentes por separado.

La alta capacidad de integración que proporciona la casa *ABE* hace que sea más rentable el incluir varios de los módulos conceptuales en que se ha dividido la solución en un solo instrumento. De este modo, la solución más económica es aquella que integra un codificador, el multiplexador y el modulador COFDM en una única unidad de *rack*. Al presupuesto detallado en [AbeP1] hay que añadirle el mezclador de frecuencia intermedia a radiofrecuencia [Abe2] cuyo precio se da en [AbeP2].

#### 2.4.2 Activa Multimedia

- Web: [www.activamultimedia.com](http://www.activamultimedia.com)
- Contacto: David Sandiu (dsandiu@activamultimedia.com, 638 874492 , 93 5672568).

*Activa Multimedia* es una empresa afincada en Barcelona que proporciona soluciones y servicios para el sector audiovisual. De hecho, es una filial de la *Corporació Catalana de Mitjans Audiovisuals* (CCMA). Es una empresa dedicada a proporcionar soluciones y servicios para el sector audiovisual y su actividad se enmarca en cuatro grandes áreas: TV digital interactiva, producción y gestión de vídeo, software de gestión de contenidos y servicios.

Sus objetivos como empresa de innovación y desarrollo de la CCMA es especializarse en la implementación de tecnologías y contenidos en el ámbito digital y multimedia. Centrarse en la búsqueda de soluciones avanzadas y su aplicación en productos y servicios innovadores para las empresas de la CCMA. Comercializar los productos y servicios propios y servir de plataforma de venta para las empresas de la CCMA, con el objetivo de financiar sus actividades.

Dentro de su actividad de televisión Digital Interactiva es capaz de ofrecer soluciones completas de interactividad para TDT, desde la conceptualización, desarrollo y edición de las aplicaciones, hasta el mantenimiento de los datos (EPG, teletexto digital, el tiempo...), así como el sistema de emisión de interactivos en cabecera de múltiplex. Dentro de este marco ha obtenido uno de sus mayores reconocimientos, siendo una de las proveedoras líder en contenidos meteorológicos.

### **Soluciones aportadas por la empresa al esquema global:**

- V/A playout and continuity
- Otros

*Activa Multimedia* presenta varias posibles soluciones para la implantación de un canal de televisión local de bajo coste, aunque algunas de ellas no están basadas en MHP y, por lo tanto, se alejan de la configuración propuesta en este documento. Sin embargo, son soluciones válidas aunque con menor funcionalidad y flexibilidad, por lo que también se han tenido en cuenta y se han presupuestado para ser ofrecidas al Ayuntamiento del Valle de Aranguren.

#### **2.4.2.1 Automatic TV (Canal Local, TDT Low Cost, o Canal Ciudadano)**

Permite la emisión de contenidos de televisión digital a bajo coste. La plataforma está pensada para la introducción de contenidos de audio y vídeo que parecen aplicaciones interactivas. En definitiva, las imágenes, textos y demás contenidos que se transmiten parecen una aplicación interactiva, aunque en realidad se trata de flujos de transporte de audio y vídeo.

La plataforma se vende integrada con el software de continuidad *Vector Box* [Vec1], lo que permite la inyección de diferentes contenidos audiovisuales, como podrían ser tramas en directo procedentes de una plataforma satelital, terrestre o contenidos almacenados en un ordenador. De este modo se puede programar una parrilla con diferentes contenidos ordenados en el tiempo, alternando entre aplicaciones propias del sistema y contenidos externos o producidos por el propio ayuntamiento.

El punto débil de esta solución es que la interactividad de los usuarios está muy limitada. De hecho, ni siquiera es posible la navegación en los contenidos que se transmiten. Por lo tanto, el PIM continuaría apareciendo de manera secuencial, sin posibilidad de acceso aleatorio a los contenidos.

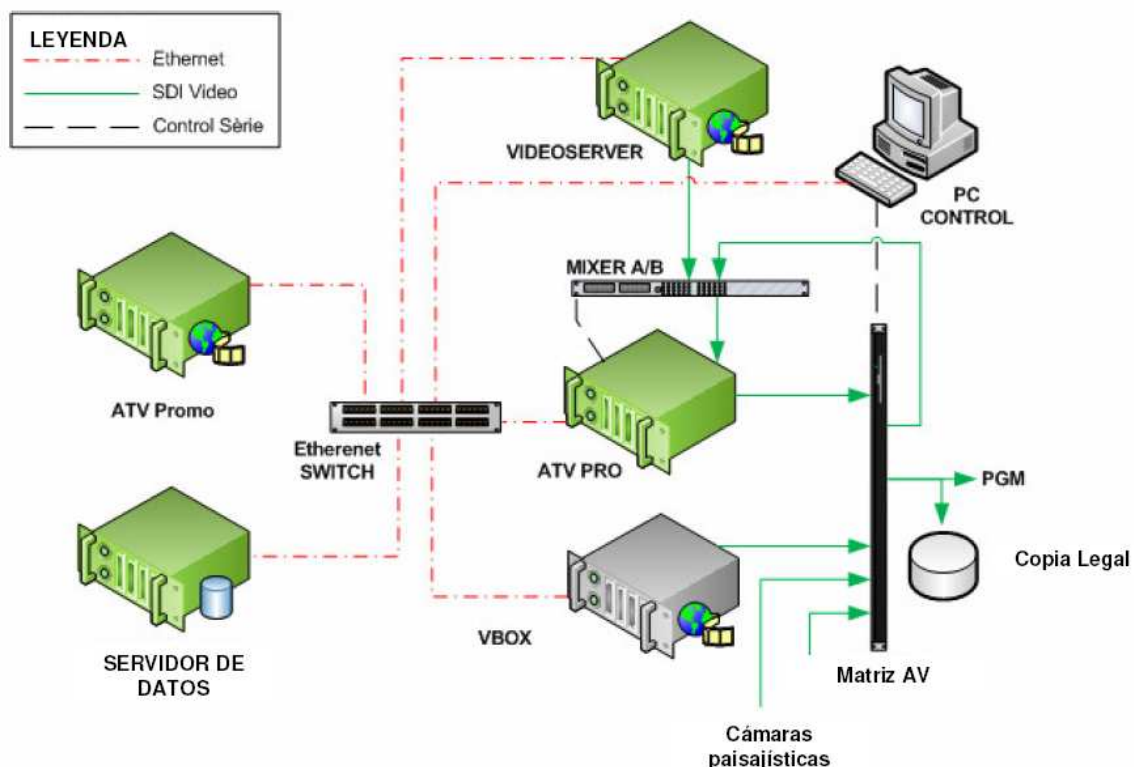
La interactividad de los usuarios se limita a contenidos que se publican en el canal de televisión, pero que son introducidos por los usuarios vía web o SMS y que son ingestados automáticamente por el sistema y convertidos en contenidos de vídeo y audio.

Los posibles contenidos a transmitir van desde noticias hasta información del tiempo, pasando por imágenes de cámaras paisajísticas o incluso pequeños vídeos subidos por los usuarios a la web (esto último requiere de un control por parte del personal responsable de la emisión).

Los contenidos son generados mediante software específico de *Activa Multimedia*, el cual permite generarlos a partir de plantillas. Es complicado realizar un diseño completo a partir de ellos, así que el sistema tiende a estancarse con el tiempo. Los software de generación de contenidos son:

- **ATV Promo.** Permite la producción automática de contenidos para TV. Dispone de varias aplicaciones: *ATV Content*, *ATV RSS*, *ATV XML*, *ATV Show Editor*.
- **ATV Pro.** Permite generar automáticamente piezas de vídeo que complementan la imagen de canal (separadores, avances, destacados, agenda ...).

En la siguiente figura se muestra el esquema básico de la plataforma:



**Figura 5: Arquitectura de la plataforma Automatic TV**

Por lo tanto, además de los componentes propios del sistema, es necesaria la compra de otros componentes externos para que la plataforma pueda ser emitida, como son:



- Software de continuidad *Vector Box*
- PC para la instalación del software de automatización de *Activa Multimedia*.
- Sistema de copia legal
- PC de control
- Matriz de vídeo

Además, existen otros componentes opcionales que mejorarían las prestaciones del sistema, como por ejemplo un mezclador para optimizar las transiciones entre programa y programa.

#### 2.4.2.2 Televisión digital interactiva

Sistema *Hand Data* que permite implantar un canal de televisión completo incluyendo aplicaciones interactivas. En concreto, permite la inserción de:

- Tablas PSI-SI
- Carruseles DSM-CC con aplicaciones MHP (AIT, Descriptores...)
- *Stream Events*
- Otros: Subtítulos DVB, PS, MPE...

Existen tres versiones de este software (*Hand Data Lab*, *Hand Data On Air Pro-Cluste* y *Hand Data On Air Lite*) pero con la más sencilla sería suficiente para los propósitos del proyecto.

Además, el sistema incluiría un repositorio de aplicaciones MHP interactivas (*iTV Suite*) incluyendo EPG, una lanzadera y aplicaciones para la inserción de mensajes de texto de manera sencilla. En cualquier caso, la funcionalidad de las mismas es bastante limitada.

Por último, *Activa Multimedia* incluye varias utilidades software para la gestión del canal de televisión. La mayoría de ellas están orientadas a la gestión administrativa.

La aplicación GREC está orientada al diseño de parrillas y escaletas, pero no parece necesaria en el sistema, teniendo en cuenta que en la configuración básica se integra el *Vector Box*.

Una descripción más detallada de ambos sistemas se puede encontrar en el propio presupuesto proporcionado por *Activa Multimedia* así como en la documentación adjunta a este documento [ActP1].

#### 2.4.3 Alcázar de San Juan

- Web: [www.fmpee.es](http://www.fmpee.es)
- Web: [www.alcazardigitaltdt.com](http://www.alcazardigitaltdt.com)
- Contacto: Juan Angel Sanchez (jasanchez@fmpee.es, 629420661).

El proyecto de la televisión de Alcázar de San Juan llamado "Alcázar Digital TDT" fue iniciado en el año 2006 y esta dentro de la iniciativa e-europa de la Unión Europea que tiene como iniciativa promover la extensión de la sociedad de la información como medio para mejorar la competitividad de las empresas y la calidad de vida de los ciudadanos.

Dentro de esta iniciativa, está el programa "Ciudades Digitales", promovido por el Ministerio de Industria, Comercio y Turismo, que pretende crear unas "islas" digitales donde experimentar sistemas, soluciones, servicios y aplicaciones, y aprender sobre comportamientos y el impacto de cada alternativa comprobada. Estas Ciudades Digitales actuarán a su vez como demostradores vivos que muestren a la sociedad en general las ventajas que se derivan de la extensión de la sociedad de la información.

El proyecto, que se denomina "Alcázar Digital TDT" pretende desplegar en una ciudad de tamaño medio un sistema interactivo universal a través del televisor y mediante el uso de la plataforma abierta MHP, con vocación de llegar a todos los ciudadanos y poner a su disposición una gama amplia de servicios. Para ello dispuso de gran apoyo tanto público (Ayuntamiento de Alcázar de San Juan, Junta de Comunidades de Castilla la Mancha, Ministerio de Interior, Comercio y Turismo, Universidad de Castilla la Mancha) como privado (Telecom Castilla la Mancha, Caja Castilla la Mancha, Microsoft...).

En cuanto a la oferta de servicios, se estructura en tres áreas:

- a) Servicios de utilidad pública, ofrecidos por Administraciones Públicas (e-Administración sobre el televisor).
- b) Servicios ofrecidos por proveedores locales (comercio, empresas de servicios, medios de comunicación) para los que se construirá una plataforma que facilite la creación y el funcionamiento de aplicaciones y para los que se realizarán acciones de estímulo que incentiven su participación.
- c) Servicios ofrecidos por empresas nacionales que quieran usar el demostrador de Alcázar de San Juan como un *tested* sobre el que evaluar las nuevas oportunidades de negocio que existen alrededor de la televisión digital.

En este caso no se trata de una empresa, sino de un ayuntamiento de Ciudad Real que desde hace muchos años tiene desplegada una televisión local y que a día de hoy es uno de los referentes en televisión digital interactiva (bajo MHP) en España. Sus instalaciones son muy completas y en cada uno de las secciones en las que hemos dividido la implantación de la televisión del Valle de Aranguren se implementan de la siguiente manera:

- V/A payout and continuity: VSN
  - Tienen instaladas las siguientes utilidades, aunque a algunas de ellas no les dan uso:
    - **VSNmatic** (software de continuidad)

- **VSNnetPlayer**
- **VSNarchive** (sistema de archivo)
- **VSNcg** (tituladora)
- **VSNlegalRec** (copia legal)
- **VSNprompter** (el prompter)
- **VSNnews** (software de edición de noticias)
- MHP *playout and management*: Alticast
  - **AltiComposer**
  - **AltiInteractive**
  - **AltiSynchro** (permite el lanzamiento de *stream events*)
- Multiplexer: Thomson
  - Thomson DBX 4300
  - Emplean tres de las entradas ASI del multiplexor para generar una única salida ASI en la que se integra:
    - V/A (trama de vídeo y audio procedente del sistema de continuidad)
    - Aplicaciones MHP
    - OTA (trama para la actualización del software del receptor). Por el momento no han actualizado el *middleware* MHP del receptor. Solamente su *firmware* para mejorar sus prestaciones.
  - Disponen de una matriz **Kramer VS-8000** de ocho entradas y ocho salidas. **VSNmatic** es capaz de controlar esa matriz, de tal forma que en un momento dado puede conmutar entre la señal a la salida del
- STB receiver
  - Los receptores empleados en Alcázar de San Juan son de la casa coreana *Handan*, con *middleware* MHP 1.02 de *Alticast*, sistema de acceso condicional CONAX y sin capacidad para decodificar alta definición.
  - La distribución de los receptores en la población se realizó mediante cesión por parte del ayuntamiento, a un coste de 30 euros y con un compromiso por parte del usuario de responder a una encuesta mensual sobre la televisión digital y sus aplicaciones. Al cabo de un año terminaba el contrato de cesión con lo que el ciudadano perdía el derecho a la devolución de los 30 euros, mientras que el ayuntamiento cesaba en la obligación de devolverlos. De esta forma se consiguió subvencionar de manera legal el acceso a los receptores.
- Otros
  - Dispone de un canal en *Youtube* a través del cual se puede tener acceso en diferido a los contenidos emitidos (<http://www.youtube.com/user/AlcazarDigitalTDT>).
  - Disponen de una licencia del software de edición **Canopus Edius NX**.
  - En la sala de control disponen de un monitor **Kaleido Alto** (de *Miranda Technologies*) para la visualización en un televisor de plasma de las diferentes fuentes de vídeo a insertar en el canal local.

#### 2.4.4 Appear TV

- Web: [www.appeartv.com](http://www.appeartv.com)
- Contacto:
  - James Chance ([james.chance@appeartv.com](mailto:james.chance@appeartv.com)).
  - Tom Erik Hagen ([tom.erik.hagen@appeartv.com](mailto:tom.erik.hagen@appeartv.com))

*AppearTv* es una empresa con sede en Oslo y fue fundada en 2005. Se dedica al diseño fiable, de generación de ingresos y soluciones innovadoras para los operadores que buscan ofrecer contenido en tiempo real a la casa.

Ofrece soluciones totalmente flexibles para los operadores de televisión digital terrestre utilizando IP o cualquier otro estándar para la transmisión de TDT de señales de vídeo y de audio.

#### Soluciones aportadas por la empresa al esquema global:

- Multiplexer
- COFDM Modulator

Esta familia de multiplexores nos fue recomendada por Johannes Schmid, de *MIT-Experts*, por lo que la consideramos una opción muy a tener en cuenta, sobretodo debido a su precio relativamente bajo en relación a las prestaciones que ofrecen [App1, AppP1].

El multiplexador **MC 3000**. [App1]. Tiene gran capacidad para tener diferentes tipos de configuraciones de entrada y salidas entre las que se podrá elegir.

- Diferentes configuraciones de entrada:
  - Entrada ASI : dispone de 3xASI mediante conectores BCN que soportan hasta 213Mbits por entrada.
  - Entrada Ethernet: 1 entrada 10/100/1000BaseT card que soporta UDP/ RTP Multicast/Unicast reception. Dispone un interfaz Mini-GBIC para entradas ópticas.
  - Entrada DVB-S/S2: dispone de 4xDVB-S/S2 entradas que soportan modos DVB-S, DVB-S2 QPSK and 8PSK.
  - Entrada DVB-T: dispone de 4xCOFDM entradas. Soporta modulaciones QPSK, 16QAM, 64QAM.

Además las cuatro configuraciones que se pueden escoger, tienen un puerto Ethernet para la gestión

- Diferentes configuraciones de salida:
  - Salida ASI: dispone de 4xASI mediante conectores BCN que soportan hasta 213Mbits por entrada. Regeneración de tablas PSI/SI.
  - Salida Ethernet: 1 entrada 10/100/1000BaseT card que soporta recepción UDP/ RTP Multicast/Unicast. Dispone un interfaz Mini-GBIC para entradas ópticas. Regeneración de tablas PSI/SI.

- Salida QAM: dispone de 8xQAM que soportan modulaciones 32 / 64 / 128 / 256 QAM. Regeneración de tablas PSI/SI.
- Salida decodificadora de video compuesto y audio analógico: dispones de 2 salidas decodificadoras.

### 2.4.5 Cinegy

- Web: [www.cinegy.com](http://www.cinegy.com)
- Contacto: La distribución en España la lleva *Promovisa* (que en realidad lleva *SIDSA*, *Dektec*, *Cinegy* y *Pebble Beach*)
  - Guillermo Moreno ([gmoreno@promovisa.es](mailto:gmoreno@promovisa.es)).
  - Tomas Nielsen ([tnielsen@promovisa.es](mailto:tnielsen@promovisa.es) )
  - Juan Limón ([jlimon@promovisa.es](mailto:jlimon@promovisa.es))
  - Troy Patterson ([troy@cinegy.com](mailto:troy@cinegy.com)).

*Cinegy* es una empresa con sede en Estados Unidos y Alemania. Centra su actividad en la investigación y el desarrollo de los medios de comunicación y el software de televisión y sus tecnologías. El desarrollo de este trabajo comenzó hace 10 años y esta basada en hardware de PC de escritorio para el *front-end* y equipos informáticos estándar de servidor en el *back-end*, todos sentados en la infraestructura de red existente. Este diseño constituye la base del flujo de trabajo que siguen hoy en día. El trabajo dio lugar a primeros productos pilotos durante el año 2001 y los primeros productos comerciales estuvieron disponibles en 2002. La plataforma *Cinegy* desde el principio fue diseñada como un sistema de nivel de empresa que es independiente de la resolución y del sistema de almacenamiento. Además, la arquitectura es extremadamente escalable y absolutamente abierta. Con el tiempo, los elementos adicionales tales como la integración de noticias y programa de emisión se han ido añadiendo a la solución completando la propuesta inicial.

Gracias a este desarrollo el producto de *Cinegy* proporciona un sistema completo que integra todas las áreas de producción de los medios de comunicación en un amplio software homogéneo.

### Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity

Aunque ellos definen su software ***Cinegy eXpress*** como una solución de broadcast, en realidad solo cumple con la parte de continuidad, gestión de contenidos y archivo [Cin2]. ***Cinegy eXpress*** está compuesto de varios elementos software diferentes los cuales pueden ser incorporados paulatinamente a un sistema, de tal forma que permiten un crecimiento paulatino en la funcionalidad de un sistema.

Todas sus herramientas software están perfectamente conectadas, disponiendo además de un software de mensajería instantánea para la comunicación entre los diferentes puestos.

Por un lado, dispone de un sistema de ingesta de contenidos (**Cinegy Ingest** [Cin4]) que permite la ingesta simultánea desde varias fuentes y a varios formatos diferentes, de tal forma que posteriormente se puede trabajar con diferentes calidades. También permite añadir meta-información al contenido ingestado para su posterior búsqueda. La ingesta se puede realizar sobre cualquier disco duro (no tienen limitaciones en ese sentido ni requerimientos hardware extraordinarios).

Dispone de un sistema de archivo (**Cinegy Archive** [Cin6]) y una base de datos (que puede ser dividida en varias bases de datos si así lo desea el usuario), lo que facilita sobremanera la gestión de todo el material audiovisual.

También es capaz de realizar ediciones rudimentarias de los contenidos, incluyendo titulaciones, transiciones, cortes y pequeños filtros (**Cinegy Desktop** [Cin5]). EL software de edición es capaz de trabajar con cualquier codec instalado en el ordenador. Además, realiza la edición en tiempo real y sin necesidad de renderizado. En cualquier caso, no sería capaz de sustituir a un software de edición tipo *Avid Lyquid*. Desde la empresa aseguran que la integración con este último software no sería difícil de realizar, aunque no está incluido en el paquete que ofertado [CinP1].

La continuidad y emisión o “*playout*” del sistema se realiza a través del software **Cinegy On Air** [Cin3]. Es un software de continuidad bastante rudimentario ya que no incluye funcionalidades básicas como podrían ser la inclusión de un vídeo en bucle para rellenar los huecos entre contenidos a transmitir. Su “*playout*” es a través de SDI, UDP o AES/EBU.

Desde *Cinegy* se han mostrado muy interesados en el proyecto del valle de Aranguren, ya que hasta la fecha no tienen ningún cliente en España y les interesa introducirse en este mercado a toda costa. Su oferta incluye por 10.000 euros, los siguientes productos, todos ellos con funcionalidad para trabajar en HD (la versión SD ya no se comercializa) [CinP1]:

- **Cinegy On Air** [Cin3]. Software para la automatización de emisión en tiempo real. Soporta tanto SD como HD con salida SDI. Soporta los formatos de video más comunes (AVI, DV, HDV, IMX, XDCAM SD/HD, MPEG2 (up to 1080i 4:2:2), AVID DNxHD, AVC-Intra 50/100, Quicktime and Windows Media). Permite control remoto vía LAN.

Tiene capacidad de insertar video en tiempo real por medio de RTP/UDP. Además permite hacer la emisión de diferente canales a la vez, incluso teniendo diferentes fuentes y formatos de salida.

- **Cinegy Ingest** [Cin4]. Software con diferentes funcionalidades dirigidas a facilitar la ingesta de video (su tratamiento y posible almacenado para trabajar con el resto del sistema). Permite hacer diferentes tipos de ingestas de video:
  - Ingesta de contenido en alta definición.
  - Ingesta DVB.
  - Ingesta simultanea en varios formatos y generación de *proxies*.

- Ingesta directa a formato AVID MXF.
  - Ingesta de meta-datos
  - Ingesta por lotes.
- **Cinegy Archive** [Cin6]. Software para gestionar de forma adecuada todos los videos de que se dispone. Permite a los miembros del equipo acceder al material desde cualquier lugar y ver los cambios que se hacen en él en tiempo real.

Permite crear diferentes derechos de acceso al material, desde un nivel mínimo hasta un acceso total.

- **Cinegy Desktop** [Cin5] (5 licencias). Es una herramienta de producción que proporciona acceso en tiempo real a los medios de cinegy **Archive** [Cin6]. Facilita el trabajo en grupo por medio de una mensajería instantánea entre los usuarios del sistema.

Ofrece todos los controles estándar para ver el material de video, tales como el control de la reproducción con el ratón, saltar a eventos, zoom ó herramientas para facilitar la búsqueda del origen del clip.

También ofrece numerosas herramientas para la generación de la línea de tiempo como insertar videos por medio de arrastrar y soltar, copiar y pegar... etc.

Otras herramientas de que dispone para mejorar el acabado final son por ejemplo: capacidad para controlar la velocidad de reproducción del video, congelado de imágenes, valores de imagen como opacidad, color...

Es importante resaltar que la gama de productos *Cinegy* es totalmente de software, pudiendo funcionar en cualquier PC con unas características mínimas nada difíciles de cumplir a día de hoy [Cin7].

#### 2.4.6 Dektec

*Dektec* es una empresa con sede en Holanda aunque tiene distribuidores por todo el mundo. Su actividad fundamental está centrada en la fabricación de componentes para difusión y tratamiento de señales de vídeo. Su extensa gama de tarjetas de PC integradas [Dek1], así como la alta fiabilidad de las mismas, la ha situado como una de las empresas más importantes del sector.

- Web: [www.dektec.com](http://www.dektec.com)
- Contacto: La distribución en España la lleva *Promovisa* (que en realidad lleva *SIDSA*, *Dektec*, *Cinegy* y *Pebble Beach*)
  - Guillermo Moreno ([gmoreno@promovisa.es](mailto:gmoreno@promovisa.es)).
  - Tomas Nielsen ([tnielsen@promovisa.es](mailto:tnielsen@promovisa.es))
  - Juan Limón ([jlimon@promovisa.es](mailto:jlimon@promovisa.es))
  - Han Roering ([Han.Roering@dektec.com](mailto:Han.Roering@dektec.com))



### Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity
- Multiplexer

El software **DTC-700 MuxExpert** permite la multiplexación de varias tramas de transporte, así como la generación de una lista de reproducción (*playlist*) [Dek5].

Puede remultiplexar varias tramas de transporte. Para ello, cada una de estas tramas tiene que ser introducida al PC, pudiendo dicho proceso ser realizado a través de tarjetas *Dektec* o a partir de ficheros en disco duro (o de ambas simultáneamente). Estas entradas pueden ser hasta 4 en caso de que se utilice alguna entrada a partir de ficheros en disco duro y hasta 8 en caso de que todas provengan de tarjetas instaladas en el PC.

La configuración de la remultiplexación se puede hacer a través de una interfaz gráfica, o mediante ficheros XML.

En cualquiera de los casos, la lista de reproducción de **DTC-700 MuxExpert** [Dek5] es muy rudimentaria (no permite hacer bucles ni parrillas genéricas) pero podría ser suficiente para las necesidades de este proyecto.

El software incorpora una API para el control del sistema, la cual podría ser empleada para implementar una escaleta de alto nivel o para controlar la configuración de la trama de transporte. Sin embargo, este proceso requiere de un tiempo de desarrollo (y un coste en personal humano) que haría aumentar el precio de esta solución hasta el punto de no hacerla rentable.

Las tarjetas **Dektec DTU-225** [Dek6] permite la recepción de tramas de vídeo digital en formato SDI, así como de tramas de transporte en formato ASI. De este modo, resulta ideal para el procesamiento de dichas señales en un PC y puede ser utilizada como interfaz de entrada para las diferentes soluciones que emplean tarjetas *Dektec* como moduladores o multiplexores.

Por otro lado, la tarjeta **Dektec DTA-105** [Dek7] permite transmitir desde un PC tramas de vídeo en formato SDI o tramas de transporte en formato ASI, siendo de este modo el complemento ideal a la **Dektec DTU-225** [Dek6] actuando como puerto de salida de la señal.

Empleando **DTG-700 MuxExpert** [Dek5] en combinación con tarjetas *Dektec* será posible generar una trama de transporte que combine una trama de transporte que contiene las aplicaciones MHP con otra que contenga el vídeo que se desea introducir. Además, este vídeo podría provenir de un sistema de continuidad cualquiera de los mencionados en esta memoria, a través de una trama SDI (empleando la tarjeta **Dektec DTU-225** [Dek6]) o a partir de una señal de satélite (empleando la tarjeta **Dektec DTA-2137** [Dek4]).

Por último, la familia de productos *Dektec* también dispone de tarjetas moduladoras COFDM:

- Modulador COFDM **Dektec DTA-110T** [Dek2]. Tarjeta de PC integrada. Toma la trama de datos directamente del PC, lo que obliga a que todo el tratamiento que se da a la señal sea vía software. Además, con esta tarjeta no sería posible la transmisión de contenidos propios en directo. Tiene 2 conectores de salida RF con modulación COFDM. Como opciones de modulación permite QPSK y 16-64 QAM y se puede utilizar con portadoras 2K, 4K ó 8K. Su nivel máximo de potencia de salida es de -29 dBm. Esta tarjeta podría ser utilizada para la transmisión en bucle de un vídeo almacenado en el ordenador, por un precio realmente bajo, lo que permitiría una rápida digitalización del contenido que se está transmitiendo a día de hoy en el canal propio del ayuntamiento.
- Modulador COFDM **Dektec DTA-112T** [Dek3]. Esta tarjeta tiene características muy similares a la anterior, pero con la ventaja añadida de que es capaz de recibir una trama de datos a través de un puerto exterior ASI. Por lo tanto, sería la opción necesaria en caso de querer transmitir contenidos en directo generados en el propio Valle de Aranguren. Como contrapartida, su potencia de salida es ligeramente inferior a la primera, alcanzando los -30.5 dBm. Por último, resaltar que esta tarjeta es capaz de modular la trama digital en diferentes formatos, tales como QPSK, QAM y COFDM.

#### 2.4.7 Eurotek

- Web: [www.eurotektel.com](http://www.eurotektel.com)
- Contacto: Paolo Massone (pmassone@eurotektel.com, teléfono?).

*Eurotek* es una empresa con sede en Rivalta Scrivia en la provincia de Alessandria, dentro de *Scientific, Technological and Telecommunication Park of Scrivia Valley*. Esta especializada en el diseño y fabricación de equipos, sistemas y accesorios para telecomunicaciones. Están presentes en diferentes campos de las comunicaciones como radio y televisión, redes *wireless* y transmisión de datos. Sus principales productos son enlaces de microondas tanto analógicos como digitales, ya sean móviles como fijos y productos para la difusión de radio y televisión (moduladores, encoders, antenas...)

#### Soluciones aportadas por la empresa al esquema global:

- Multiplexer
- COFDM Modulator

La familia de productos *Eurotek* está basada en un sistema operativo Linux embebido en un módulo para *rack*, al cual se pueden añadir diferentes funcionalidades, desde codificadores hasta moduladores, pasando por multiplexadores. En definitiva, su sistema podría integrarse en el esquema global de varias formas:

- Multiplexador **EK-AMX/4** [EurP1, Eur2]. Dispone de 4 entradas ASI, que se multiplexan en una salida ASI. El Máximo *bit rate* de salida que puede soportar es de 215Mbps. Dos modos de funcionamiento, manual y auto. En el auto se regenera las tablas PSI/SI de forma automática. El modo manual permite un control total de la multiplexación al usuario.
- Multiplexador **EK-AMX/4** + Modulador **EK-DTX/1**. [Eur1] Dispone de cuatro entradas ASI, una para las aplicaciones MHP y otra para el contenido de vídeo y audio, una salida de RF modulada en COFDM, que permite como opciones de modulación QPSK y 16-64 QAM y se pueden utilizar con portadoras 2K, 4K ó 8k. La salida del modulador solo puede ser de 50 ohmios, lo que reduce considerablemente la potencia a transmitir, esta podrá ser de entre -8 a +12dBm.
- Codificador **EK-ENS/2** + Multiplexador **EK-AMX/4** + Modulador **EK-DTX/1** [Eur1]. Dispone de una entrada ASI y otra entrada SDI (con audio embebido) y una salida de RF modulada en COFDM. Las opciones de modulación y multiplexación son las explicadas anteriormente.

El sistema permite guardar configuraciones, así como programar en el tiempo un cambio de configuración. Esto permite conmutar entre diferentes configuraciones aunque esta funcionalidad no es necesaria en el caso que nos ocupa.

*Eurotek* pide que se especifique exactamente el *bit rate* de cada una de las entradas y salidas para nuestro sistema, ya que en función de ello el precio puede variar. En caso de optar por esta empresa, habría que tener mucho cuidado en la especificación final de todo el sistema. Además, no es excesivamente barata, por lo que parece más sensato optar por otra opción.

## 2.4.8 Grassvalley Thomson

- Web: [www.thomsongrassvalley.com](http://www.thomsongrassvalley.com)
- Contacto:
  - Le Doeuff Eric ([Eric.Le-Doeuff@grassvalley.com](mailto:Eric.Le-Doeuff@grassvalley.com)).
  - Fernando Polanco ([fernando.polanco@grassvalley.com](mailto:fernando.polanco@grassvalley.com))
  - Mikel Darrigues ([Mikel.Darrigues@grassvalley.com](mailto:Mikel.Darrigues@grassvalley.com), 607 203807, 91 5120358)

*Grassvalley* es una empresa con sede en California. Esta dentro del grupo *Thomson*, lo cual le hace tener puntos de distribución por todo el mundo. *Thomson* nació en 1893 y desde el principio estuvo inmerso en el proceso de imágenes en movimiento. No fue hasta 1958 cuando se creó *Grassvalley*, empresa especializada en la emisión de televisión, lo cual le hace ser una de las empresas con más experiencia en el sector. Además en 1974 se fusionó con *Tektronix* para así formar una de las empresas más potentes del sector.

### Soluciones aportadas por la empresa al esquema global:

- MHP playout and management

- Multiplexer

Tiene dos productos muy interesantes para el este proyecto, un multiplexador y un servidor de aplicaciones MHP.

- Multiplexor **Amber II Starter** [Gra1]: Es un multiplexador muy sencillo que en su forma mas básica tiene solo 2 entradas, una para el video y otra para la aplicación. El *bit rate* que soporta es de 180Mbps de entrada/salida, además es capaz de monitorizar el *bit rate* para optimizar mejor el ancho de banda. Permite gestionar completamente la generación de las tablas PSI/SI.
- **Coral MHP Server**: Servidor de aplicaciones MHP. Además de permitir el control total sobre tablas de información y servicio, *descriptores* MHP etc. Existen dos versiones:
  - **Coral TNM-5301-EL** [Gra2] Permite la gestión total de la emisión de aplicaciones MHP además de permitir la generación completa de la guía electrónica de programas (EGP). Las versiones de MHP que soporta son MHP 1.0, 1.1 y 1.2. Tiene una gran capacidad para la optimización del ancho de banda. Permite el control remoto del equipo mediante el protocolo SNMP.
  - **Coral TNM-5301** [Gra2]: Su funcionalidad es idéntica al anterior, pero dispone de una entrada ASI adicional a través de la cual es posible inyectar una trama de transporte de audio y vídeo a multiplexar con la aplicación. De este modo, se puede prescindir del multiplexor en el caso de querer transmitir una única trama de audio y vídeo. Sin embargo su precio es mucho más elevado que el del anterior, por lo que realmente no compensa.

#### 2.4.9 Icareus

- Web: [www.icareus.com](http://www.icareus.com)
- Contacto:
  - Toni Leiponen (toni.leiponen@icareus.com).
  - Nuutti Rautiainen (nuutti.rautiainen@icareus.com)

*Icareus* es una empresa afincada en Helsinki (Finlandia). Fue fundada en 2001 y desde el principio centro su actividad en la televisión digital interactiva, siendo a día de hoy una de las empresas líderes del sector, lo que le ha llevado a abrir sedes en Estados Unidos, México y China. En la actualidad los sistemas de *Icareus* están instalados en 20 países del mundo, sistemas que dan una solución completa al problema de la emisión de televisión digital interactiva, lo que le hace ser una empresa de enorme potencial.

#### Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity
- MHP playout and management
- Multiplexer

*Icareus* dispone de varias plataformas con diferente funcionalidad para la gestión y manejo de canales de televisión digital interactiva.

La solución más sencilla y que cumple perfectamente con los requerimientos del canal local que se pretende echar a andar en el Valle de Aranguren está basada en dos equipos: un servidor de vídeo y un servidor de aplicaciones MHP que integra un multiplexor.

- ***Icareus Playout VS-100 Video Server*** [Ica9]. Es un servidor de vídeo que permite hacer escaletas mediante un reproductor al que se añaden los vídeos a transmitir de manera secuencial, de tal forma que al finalizar el anterior comienza el siguiente (FIFO). Además permite añadir un video de fondo que complete los huecos de la emisión. El equipo permite crear la guía electrónica de programas (EGP), y genera de forma automática las tablas PSI/SI. Como funcionalidades añadidas (no incluidas en la oferta presentada), el software permite trabajar con tramas externas en formato SDI, S-Video y vídeo compuesto. Dispone de dos salidas ASI idénticas implementadas mediante una tarjeta *Dektec DTU-105*.
- ***Icareus Playout PC100 Compact*** [Ica6]. Es un servidor de aplicaciones MHP funcionalidad para optimizar el *object carrusel*, generación de tablas PSI/SI, planificador de para la ingesta de las aplicaciones y dispone de un codificador para la ingesta de video en tiempo real. La salida es ASI, también implementada mediante una tarjeta *Dektec DTU-105*.

#### 2.4.10 Kramer Electronics

- Web: [www.kramerspain.com](http://www.kramerspain.com)
- Contacto: Eusebio Díaz (info@kramerspain.com, 91 7478410).

*Kramer Electronics* es una empresa con sede en Estados Unidos pero después de casi 30 años de recorrido tiene puntos de distribución en todo el mundo. *Kramer* ha estado desarrollando soluciones en video, audio y gestión de señales informáticas. Con fuertes inversiones en investigación y el desarrollo, *Kramer* introdujo algunos de los "primeros" procesadores de señal de la industria. Es una empresa potente lo que le ha llevado a tener una red de con oficinas y almacenes de ventas alrededor del todo mundo.

#### Soluciones aportadas por la empresa al esquema global:

- Otros: matriz de conmutación

*Kramer* es la empresa referente a nivel mundial en matrices de conmutación de vídeo, tanto analógico como digital.

Dada la pequeña magnitud del proyecto, se requiere una matriz de conmutación SDI lo más sencilla posible (con el menor número de entradas y salidas) pero que permita trabajar con vídeo en alta definición de tal modo que el sistema no quede obsoleto en caso de que en un futuro se quiera incorporar esta funcionalidad. Además, es necesario que la matriz

sea controlable a través del puerto RS-232, de tal forma que las plataformas de continuidad puedan realizar su labor de manera automática. La matriz de conmutación SDI *Kramer* con las prestaciones mínimas para cumplir con las especificaciones anteriormente descritas es la **Kramer VS-41-HD** [Kra1].

#### 2.4.11 Mainconcept

- Web: [www.mainconcept.com](http://www.mainconcept.com)
- Contacto: [sales@mainconcept.com](mailto:sales@mainconcept.com)

Empresa fundada a principios de 1993 en *Aachen* (Alemania). En sus inicios, la compañía se especializó en software de edición de vídeo para diversos sistemas informáticos, tales como Amiga, OS/2, y Windows 95. *MainConcept* lanzó su primer *codec* de vídeo, *Motion JPEG*, en 1995 y tuvo un gran éxito así que desde entonces, la empresa se centro más en la prestación de los *codecs* de vídeo de alto rendimiento y las tecnologías asociadas (multiplexación, *streaming*, *codecs* de audio, etc).

A largo de los años ha venido desarrollando una gama completa de estándares de *codecs*, lo que le ha llevado a establecer relaciones a largo plazo con numerosas empresas líderes en el mundo, incluyendo *Adobe*, *Leitch / Harris*, *Microsoft*, *MobiTV*, *Sonic Solutions*, *Sony* y *Panasonic*.

Para fortalecer aún más la empresa y su alcance tecnológico, *MainConcept* busco un socio tecnológico con un producto sinérgico, lo que le llevo en Noviembre de 2007 a convertir la compañía es una subsidiaria *DivX, Inc*.

A día de hoy es, sin duda, una empresa referente en su sector.

#### Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity

*Mainconcept Reference*, es un software para codificación y decodificación de archivos de audio y vídeo [Mai1]. Su precio depende del número de *codecs* que se instale en el paquete de software pudiendo llegar a ser prohibitivamente caro. Sin embargo, para el propósito del presente proyecto solo es necesario que sea capaz de codificar a *MPEG-2* en definición estándar, resultando de este modo relativamente asequible [MaiP1]. Además, el software permite la incorporación de nuevos *codecs* en un futuro. En particular, la extensión para comprimir contenido en alta definición sería prácticamente inmediata.

#### 2.4.12 MIT-xperts

- Web: [www.mit-xperts.com](http://www.mit-xperts.com)
- Contacto: Johannes Schmid ([schmid@mit-xperts.com](mailto:schmid@mit-xperts.com)).

*MIT-xperts* es una empresa con sede en Munich, Alemania y fue fundada en 2001. Está especializada en desarrollo de software y consultoría en DVB (*Digital Video Broadcasting*) y MHP (*Multimedia Home Platform* / Televisión Interactiva). *MIT-eXperts* desde el principio ha intentado implementar soluciones específicas para las necesidades de los clientes, manteniendo así una estrecha relación con los mismos, lo que la hace una buena solución para pequeñas televisiones o centros tecnológicos.

#### Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity
- MHP playout and management
- Multiplexer
- COFDM Modulator

#### 2.4.13 OMB

- Web: [www.omb.com](http://www.omb.com)
- Contacto: Iván Perallón ([ivan@omb.com](mailto:ivan@omb.com), +34 976 141717).

*OMB Sistemas electrónicos S.A.* es una compañía nacida hace más de 20 años como continuación de la empresa *Electrónica Ormad* formada en 1964 y dedicada a la fabricación de retransmisores de TV.

*OMB* es una empresa especializada en la fabricación de Sistemas Radiantes, Transmisores de Radio y Transmisores de TV. Es posible encontrar productos OMB en países tan exigentes tecnológicamente y desarrollados como Estados Unidos, o en países con condiciones ambientales y técnicas extremas como Ghana, en el continente africano.

#### Soluciones aportadas por la empresa al esquema global:

- Multiplexer
- COFDM Modulator

*OMB* dispone de moduladores y codificadores integrados en una única unidad de *rack*.

En concreto, el producto más interesante es el **MOT 5 DVB-T Compact** [Omb1] el cual parece estar escalado exactamente para las dimensiones del proyecto en el Valle de Aranguren. Incluye:

- Dos entradas divididas en tres puertos:
  - Entrada de la señal de vídeo local:
    - SDI para el vídeo
    - Audio: entrada analógica



- Entrada TS externo:
  - *DVB-ASI*.
- El instrumento incluye un codificador en tiempo real para codificar a *MPEG-2* la señal de vídeo que se introduce por el puerto SDI.
- Finalmente, la trama multiplexada es modulada en COFDM. La señal modulada se inyecta a través de un puerto a 50 ohmios y con 5 W de potencia.

Entre los posibles inconvenientes del dispositivo hay que destacar que:

- Nunca ha sido probado con una trama de transporte MHP.
- No dispone de una interfaz gráfica, controlándose por línea de comandos a través de un puerto RS-232.
- No es posible programar configuraciones para que varíen en el tiempo.
- Requiere de una entrada externa de audio analógico, lo que reduce la calidad de la señal final que se transmite al obligar a una doble conversión de digital a analógico y de analógico a digital (para ello sería posible usar los componentes de la casa *Aja* ([www.aja.com](http://www.aja.com))).
- Su salida es de 50 ohmios, por lo que se requiere un adaptador que haría que la potencia donada por el equipo fuera sustancialmente menor.

#### 2.4.14 Pebble Beach

- Web: [www.pebble.tv](http://www.pebble.tv)
- Contacto: La distribución en España la lleva *Promovisa* (que en realidad lleva *SIDSA*, *Dektec*, *Cinegy* y *Pebble Beach*)
  - Guillermo Moreno ([gmoreno@promovisa.es](mailto:gmoreno@promovisa.es)).
  - Tomas Nielsen ([tnielsen@promovisa.es](mailto:tnielsen@promovisa.es))
  - Juan Limón ([jlimon@promovisa.es](mailto:jlimon@promovisa.es))
  - Rafael Dubois ([rafael@pebble.tv](mailto:rafael@pebble.tv))

*Pebble Beach* es una empresa con sede en la ciudad de *Weybridge* en el Reino Unido. Se fundó en el año 2000 y desde su inicio se centró en buscar soluciones a la automatización de emisiones de televisión. Los sistemas *Pebble Beach* se caracterizan por tener una gran flexibilidad para apoyarse en una amplia gama de aplicaciones de emisión. Diseñado alrededor de una arquitectura modular y escalable, los productos *Pebble Beach* de automatización de sistemas son altamente configurables y robustos.

Sus productos han tenido una gran acogida en el mercado, lo que le ha llevado a un gran crecimiento de la empresa, habiendo abierto también sedes en Estados Unidos y Oriente Medio.

#### Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity

**Neptune Lite Automation**[Peb1]. **Neptune Lite** es una solución de automatización que ha sido diseñada para satisfacer las necesidades de

pequeños canales. Proporciona un marco preciso, fiable y fácil de usar que además se puede utilizar para la emisión de hasta cuatro canales de televisión.

Tiene dos herramientas básicas, **Ingest Clients** que proporciona una serie de herramientas para preparar el material que se va a emitir. Permite crear una base de datos sobre los video, con puntos de inicio y final, identificadores, duración... de forma que se facilite el trabajo a la hora de preparar una escaleta de emisión. La otra herramienta es **Transmission Client** que son todas las funciones para lista de emisión. El elemento principal es la línea de tiempo que esta diseñada para poder ver la lista de emisión de forma cómoda, pudiendo así corregir, añadir, quitar los elementos de ella.

El sistema permite ofrecer una doble redundancia entre los dispositivos controladores y el servidor lo cual da una mayor tolerancia ante fallos

#### 2.4.15 Promax

- Web: [www.promax.es](http://www.promax.es)
- Contacto:
  - Jordi Pallares (jpallares@promax.es)
  - J.M. Vilella (jmvilella@promax.es).

*Promax* fue fundada en 1963 en Barcelona. Los primeros instrumentos desarrollados por *Promax* incluían equipos para generar señales de televisión y radio y analizadores para comprobar la calidad de la recepción.

Hoy, *Promax* es una empresa líder en el sector de las tecnologías de información y comunicaciones. La empresa invierte el 15% de su volumen de ventas en Investigación y Desarrollo para mantenerse al día en un mercado tan dinámico.

Los productos incluyen una amplia gama de instrumentos de medida y especialmente para telecomunicaciones, proporcionando soluciones de medida para sistemas de cable, satélite y televisión digital terrestre.

En la actualidad dispone de una extensa red de ventas directa e indirecta por todo el mundo.

#### Soluciones aportadas por la empresa al esquema global:

- COFDM Modulator
- Otros (transmodulador analógico a digital)

Para el propósito de la modulación de la señal a transmitir por la red de cable del valle de aranguren, podemos optar por dos gamas de productos *Promax*: por un lado instrumental destinado a cabeceras con estructura tipo *rack*, y por otro instrumental destinado a un único bastidor (*Promax* llama a esta última gama de productor DTTV [Pro6]).

La solución en rack es tan sencilla como adquirir un modulador **Promax MO 160** [Pro4]. Este es un modulador COFDM de los más baratos del mercado y está testado en televisiones que están emitiendo de manera continua. De hecho, desde *Promax* comentan que en algunos casos (por ejemplo en Almansa) su modulador se utiliza para transmitir señales COFDM a través de una red de cable, tal y como se desea hacer en el Valle de Aranguren.

La segunda solución es ligeramente más barata, pero su funcionalidad es menos flexible que la primera. En este caso, además de un modulador con entrada ASI y salida COFDM para bastidor (concretamente el modelo **Promax DT-1010** [Pro1]) que es un elemento tremendamente económico, es necesario adquirir una unidad de control y fuente de alimentación (**Promax DT-800** [Pro7]), así como una estructura para *rack* y pared en la que se instala el dispositivo (**Promax DT-900** [Pro8]). Estos dos últimos elementos podrían utilizarse para instalar otros dispositivos de la misma casa, como transmoduladores DVB-S a COFDM (tipo **Promax DT-202** [Pro9]) que probablemente será necesario adquirir en el proceso de digitalización de los canales provenientes de satélite y que a día de hoy se introducen en la red en formato analógico.

Esta segunda opción no permitiría el cambio en la frecuencia de modulación del canal local, pero por lo demás es perfectamente válida. En caso de querer aprovechar la estructura **Promax DT-900** [Pro8] para este último fin, el ahorro podría ser sustancial. En cualquier otro caso, la diferencia de precio no compensa la pérdida de funcionalidad.

Finalmente, *Promax* dispone de un dispositivo capaz de modular en COFDM una señal analógica de vídeo. El **Promax DT-504** [Pro3] sería el elemento necesario en caso de adoptar la solución propuesta en el apartado 2.3.5 de este documento, en la que simplemente se digitalizaba la señal analógica que se emite en la actualidad.

#### 2.4.16 Sapec

- Web: [www.sapec.es](http://www.sapec.es)
- Contacto:
  - José Manuel López (jmlopez@sapec.es, 636 969115).
  - Rubén Martínez González (rmartinez@sapec.es)

*SAPEC* es una empresa que se fundó en 1976 en Madrid. Su propósito inicial fue dedicarse al Desarrollo, Fabricación e Instalación de Hilo Musical y Filodifusión.

A finales de la década de los 90, *SAPEC* realizó una fuerte inversión de capital en el Departamento de I+D, el objetivo fue crear nuevos productos, para ofrecer de esta forma, al mercado de las telecomunicaciones, productos de alta calidad y totalmente competitivos.

De esta forma, *SAPEC*, se especializa en Sistemas de Compresión MPEG, ofreciendo un servicio integral en todos sus productos; desde su definición,

hasta la puesta en marcha del sistema, pasando por las distintas fases de Integración e Instalación.

### Soluciones aportadas por la empresa al esquema global:

- Multiplexer
- Otros (codificador *MPEG-2*)

Multiplexor **Sivac MTS1000-D4A** [Sap1]. Multiplexor que dispone de 4 entradas *DVB-ASI* y 2 salidas *ASI* (una de ellas de monitorización). Genera las tablas *PSI/SI* además de permitir la gestión manual de ellas.

El sistema de gestión del sistema es vía web por medio del protocolo *SNMP* y también dispone de un sistema de control integrado *SIVAC-NMS*.

Codificador **Sivac 3000 MVE3010-D4A** [Sap2]. El sistema dispone de 2 entradas una digital *SDI* y otra analógica y 2 salidas *DVB-ASI*. El codificador de video que utiliza *MPEG2 MP @ ML (4:2:0)* de 2,5 hasta 50 Mbps.

El sistema de gestión es vía web haciendo la configuración muy intuitiva.

### 2.4.17 SIDA

- Web: [www.sidsa.com](http://www.sidsa.com)
- Contacto: La distribución en España la lleva *Promovisa* (que en realidad lleva *SIDSA*, *Dektec*, *Cinegy* y *Pebble Beach*)
  - Guillermo Moreno ([gmoreno@promovisa.es](mailto:gmoreno@promovisa.es)).
  - Tomas Nielsen ([tnielsen@promovisa.es](mailto:tnielsen@promovisa.es))
  - Juan Limón ([jlimon@promovisa.es](mailto:jlimon@promovisa.es))

*SIDSA* es una empresa con sede en Madrid que fue fundada en 1992. Hoy en día la empresa a tenido una gran expansión comercial en todo el mundo, y tiene oficinas en Dubai, Hong Kong, Moscú y San Francisco.

En la actualidad *SIDSA* es uno de los proveedores de tecnología líderes en el campo de *Digital Video Broadcast*, habiendo sido premiada con diferentes galardones como el *Red Herring Europe 2008*, un premio que se otorga a las 100 compañías privadas de tecnología con sede en la región de EMEA de cada año, el premio a la mejor PYME innovadora 2006 dentro del plan Avanza y otros muchos.

### Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity

*SIDSA* dispone de una solución integral para televisión digital por cable o terrestre. Sin embargo, para las soluciones propuestas para la red del Valle de Aranguren, solo resulta interesante el codificador de vídeo, ya que la forma en que los diferentes elementos se interconectan no encaja con el esquema general propuesto.

El codificador **Karina Flex** [Sid1]. Codificador que soporta los siguientes formatos de vídeo de entrada: PAL y NTSC, Video compuesto CVBS o s-videoY/C analogico y SDI digital. Soporta resoluciones de 720p horizontal y 576p vertical. El estándar de codificación que utiliza es el *MPEG-2* ISO 13818 MP@ML y *MPEG-1* ISO 11172-2. A la salida se obtiene *MPEG-2* TS sobre UDP/RTP.

## 2.4.18 Stream Labs

- Web: [www.streamlabs.es](http://www.streamlabs.es)
- Contacto:
  - Ricardo Vazquez ([rvazquez@adinet.com.uy](mailto:rvazquez@adinet.com.uy)).
  - Tho de Vietex Media ([xtn@vietexmedia.com](mailto:xtn@vietexmedia.com))
  - Maxim Mazurenko ([maxim.mazurenko@stream-labs.com](mailto:maxim.mazurenko@stream-labs.com))
  - Alexander Lunev ([la@streamlabs.ru](mailto:la@streamlabs.ru))

Es una empresa Rusa fundada en 1991. En la actualidad tiene sedes en Uruguay, Alemania y Taiwán. Esta especializada en tres áreas clave:

- Sistemas informáticos de televisión
- Sistemas de vídeo-vigilancia
- Dispositivos de monitorización *wireless*

En el campo de los sistemas para televisión, ofrecen soluciones para una gran gama de televisiones, desde pequeñas televisiones locales por cable, hasta grandes cadenas nacionales.

El hardware y software para televisión profesional de *Stream Labs* soluciona las siguientes tareas:

- Monitorización de señal de video y audio.
- Emisión de gráficos.
- Automatización de emisión.
- Diseño de canales de música, informativos, noticias y deportes.

Los equipamientos de *Stream Labs* son utilizados en más de 200 canales de TV alrededor del mundo.

### Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity

El software de continuidad **Stream Labs TeleInfo** [Str1, Str2]. El software provee automatización de la emisión mediante la inserción en una escaleta de los programas a emitir. Es capaz de agregar títulos, superponer logotipos fijos o animados, generar relojes... etc.

La aplicación no requiere ningún equipamiento adicional (sincronizadores, mezcladores, transcodificadores, sistemas de edición no lineal, etc.) y

permite ejecutar información gráfica, superponer video y audio directamente sobre la señal que atraviesa la placa.

La escaleta editable permite una gran variedad de posibilidades de edición, como reproducción cíclica de un elemento de la parrilla o la transición rápida entre eventos, entre otros.

Soporta un gran número de formatos de entrada. Además dispones de una SDK para el control avanzado de las funciones.

**Stream Labs TeleInfo** controla el *playout* que se realiza físicamente a través de tarjetas **Stream Labs** instaladas en un PC convencional [StrP1]. En función de la tarjeta que se seleccione la funcionalidad del sistema puede ser mayor o menor. En concreto, se proponen dos alternativas.

- **Stream Labs Stream Wind SDI** [Str5]. Esta tarjeta permite la emisión de contenido almacenado en disco duro, a través de una salida SDI.
- **Stream Labs Stream Wind SDI II** [Str6]. La ventaja fundamental de esta tarjeta sobre la primera es que dispone de una entrada de vídeo SDI, por lo que si en un momento dado se pretende realizar emisiones en directo o inyectar la señal proveniente de una fuente externa (de un satélite, por ejemplo) esto sería posible.

#### 2.4.19 Tandberg

- Web: [www.tandbergtv.com](http://www.tandbergtv.com)
- Contacto: Jaime Hervada (jaime.herbada@mome.es, 699941243), de la empresa MOME (<http://www.mome.es>).

**Tandberg** es una empresa con sede en Atlanta (Estados Unidos), aunque también dispones de sedes secundarias en Reino Unido y Hong kong. En 2007 fue adquirida por el grupo *Ericsson*, entrando a formar parte de la división en cargada de dar soluciones de televisión, lo que le ha dotado de sedes de distribución por todo el mundo.

Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity
- Otros

En este caso estamos interesados en uno de los codificadores *MPEG-2* ofertados por la empresa. En concreto, el modelo **E5710 MPEG-2 SD Encoder** [Tan1] el cual permite la codificación en definición estándar, siendo un instrumento robusto y que ofrece garantías. A su entrada, permite video compuesto o vídeo SDI. Además, puede codificar a *bit rate* variable ó constante.

El mayor inconveniente del equipo proviene de su elevado coste [TanP1], ya que se trata de un instrumento destinado a grandes cadenas.

#### 2.4.20 Tmira

- Web: [www.tmira.com](http://www.tmira.com)
- Contacto: Eduardo García (egarcia@tmira.com, 606512900).

*Tmira* es una empresa joven de Castilla-La Mancha que empieza su andadura a finales de 2005 a raíz de la obtención del Primer premio en el concurso de Ideas Innovadoras de Base Tecnológica de Castilla-La Mancha por su Sistema de Gestión Integral de Servicios Interactivos para Televisión Digital. Además de las oficinas en Cuenca también las tiene en Madrid. Es una compañía especializada en el desarrollo de sistemas y servicios para la televisión digital.

La tecnología de *Tmira* está basada en diferentes estándares y permite la reutilización de los contenidos ya existentes para que su puesta en marcha no suponga un esfuerzo adicional a las cadenas de televisión.

Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity
- MHP playout and management
- Multiplexer
- Otros

*Tmira* propone dos posibles soluciones para la implantación del canal de televisión digital en el Valle de Aranguren: la primera basada en MHP y receptores COFDM y la segunda basada en receptores híbridos de cable-iPTV.

##### 2.4.20.1 Solucion 1: MHP and COFDM

La solución de *Tmira* para la transmisión de aplicaciones interactivas MHP es una solución con garantías, habiendo sido demostrada y estando en funcionamiento en cadenas de televisión muy importantes, como por ejemplo RTVA (Radio Televisión de Andalucía).

**TmBroadcaster** [Tmi1]. Permite la inserción de las aplicaciones MHP, incluyendo vídeo y audio. El software se integra con:

- Inyectores ASI (*Dektec* y *Alitónica*, por el momento)
- Software de continuidad (*VSNmatic* es el que suelen usar habitualmente, pero cualquier otro puede servir)

Esquemáticamente:

- Ventajas
  - Recibe vídeo vía ASI o en un puerto IP vía UDP



- Tiene *stream events*
- Application *scheduling*
- Desventajas
  - No permite hacer la escaleta. Esta habría que subcontractarla a VSN o a cualquier otro proveedor. No es necesario que la señalización a partir del servidor de vídeo sea muy compleja, porque el propio *TmBroadcast* es capaz de generar toda la señalización y todas las tablas.

**TmInteractive** [Tmi2]. Es la parte del sistema que se encarga de la interacción con las aplicaciones que requieran de canal de retorno. Está muy bien pensado. Es un software que se encarga de recoger la información de las fuentes en las que se tenga interés (páginas web, bases de datos, RSS...) y de hacerlas accesibles de manera sencilla para las aplicaciones.

- Ventajas
  - Tiene una API de tal forma que el cliente no está "atado" a *Tmira* para toda la vida, pudiendo hacer desarrollos posteriores.
  - Sistema ya testado en otras televisiones, como RTVA: es seguro y robusto.

#### 2.4.20.2 Solucion 2. Receptores híbridos cable-IPTV

*Tmira* trabaja con varias empresas que instalan redes de cable en hoteles y pequeños pueblos. En esos casos, lo habitual es usar un receptor híbrido de cable e IPTV de tal forma que la señal de vídeo viene por la red de cable y la interactividad está basada en el canal de retorno, incluyendo *video on demand*.

Receptores que están usando:

- *A-DHT* IC8935 [Adh1]
- *Middleware*: ANT Galio

La ventaja fundamental del sistema es la posibilidad de transmitir vídeo sobre IP. Además, el propio receptor integraría el canal de retorno.

Desventajas

- No es del todo evidente que las aplicaciones se vayan a desarrollar de manera tan sencilla como en MHP y además se estaría optando por una tecnología con la que no se ha trabajado hasta ahora ni en la UPNA ni en los CES.
- No es evidente que ANT sea más potente que MHP, a pesar de que la empresa china *A-DHT* que fabrica los receptores indique en sus especificaciones que el receptor integra un HTML *browser* y otras funcionalidades avanzadas.
- El sistema de acceso condicional que pueden incorporar (según su propia hoja característica) no es ninguno de los dos grandes estándares (*Nagra* ni *Conax*).

### 2.4.20.3 Otras utilidades

Además de las herramientas anteriores, *Tmira* dispone de otras utilidades como son:

- **TmEPG** [TmiP1]. Genera una EPG. Lo puede hacer recibiendo la información desde diferentes fuentes, bien sea de manera automática a insertándola en modo manual (la Televisión de Andalucía utiliza la pestaña de *TmBroadcast* para introducir manualmente los datos). Los comerciales de *Tmira* insisten mucho en esto: es como si fuera la aplicación estrella de la compañía!
- **TmPlayer** [TmiP1]. Es una especie de navegador que recibe la aplicación en formato XML y lo traduce a MHP (o a varios otros formatos, entre ellos GEM, Ginga, ANT Galio o Oregon).
- **TmDeveloper** [TmiP1]. Una versión para pequeños desarrolladores que incluye un modulador y la capacidad de inyectar vídeo y audio además de las aplicaciones. Ahora mismo ya no tiene mucho sentido porque el propio *TmBroadcast* también permite esa posibilidad. El vídeo lo recibe vía IP (una de las opciones de *TmBroadcast*).
- **TmSubtitling** [TmiP1]. Permite insertar subtítulos en el *stream*. El sistema, tal y como aparece en la web, parece bastante avanzado porque el generador de subtítulos necesita recibir el *stream* de vídeo con los PCRs para poder sincronizar correctamente.

### 2.4.21 TVMaker (EpsilonMedia, Vision Robotics)

- Web: [www.epsilonmedia.es](http://www.epsilonmedia.es)
- [www.vision-robotics.com](http://www.vision-robotics.com)
- Contacto: Javier Ballesteros ([javier.ballesteros@epsilonmedia.es](mailto:javier.ballesteros@epsilonmedia.es), 692 736747).

*EpsilonMedia* es una empresa con sede en Madrid que lleva ya 10 años en el sector. Está especializada en la fabricación y distribución de soluciones IP orientas a mercados audiovisuales, telecomunicación y seguridad.

*Vision Robotics* es una empresa con sede en Barcelona que nació en 2004. Esta compañía desarrolla soluciones de alta tecnología para los mercados *Narrowcasting (Digital Signage)* y *Broadcasting (TV convencional)*. El producto *TvMaker* ha sido desarrollado por *Vision Robotics* aunque su distribución corre ha cargo de *EpsilonMedia* entre otras.

Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity

A pesar de que *Epsilon Media* dice tener una plataforma (***TVmaker TDT Suite*** [Tvm1]) cuyos módulos pueden satisfacer plenamente las necesidades de una cabecera de televisión digital, la realidad es que su funcionalidad se limita a la parte del la generación, gestión y transmisión de

contenidos en banda base, bien sea en formato digital (SDI) o en formato analógico.

**TVmaker TDT Suite** [Tvm1]. Este producto se encarga de dar solución a la continuidad de la emisión de televisión. Es software modular, donde los más destacados son:

- **Ingesta:** Se encarga de la captura de vídeo multiformato.
- **Playout:** Automatización de continuidad con funciones EPG/SI.
- **Archivo:** Catalogación y almacenamiento de contenidos en entorno Web.
- **Grafismo:** Grafismo dinámico avanzado (*Chyron*).
- **Publicador Web:** Portal Web de la TV con vídeo a la carta y TV en directo por streaming.
- **Monitorización:** Servicio de monitorización 24/7 con detección remota de incidencias, sistema de envío de avisos, mantenimiento y soporte al usuario.
- **Gestión de Publicidad:** Sistema de gestión de contrataciones y emisión de publicidad.
- **Playout Remoto:** Gestión remota de operación del *playout* y el grafismo.

Entrando en la parte destinada a la distribución de la señal en una red de cable, *Epsilon Media* ofrece productos de la casa *Anevia* destinados a redes de distribución por cable en hoteles o pequeñas comunidades, con funcionalidades como la redistribución de una serie de canales en emisión en la red de cable (**Flamingo 660 series** [Tvm3]) o la distribución de vídeo bajo demanda sobre IP (**Toucan 100** [Tvm2]). Sin embargo, el ámbito del proyecto en el valle de Aranguren es totalmente diferente a este así que se descarta la opción.

Además, ha sido imposible conseguir un presupuesto medianamente serio por parte de esta empresa, por lo que no se recomienda en ningún caso.

#### 2.4.22 Vector3 (Vector Box)

- Web: [www.vector3.es](http://www.vector3.es)
- Contacto: ¿? (eva.c@vector3.es, teléfono¿?).

*Vector3* es una empresa con sede en Barcelona que fue fundada en 1984 con el objetivo de desarrollar aplicaciones gráficas para ordenador. Muy pronto vieron que donde más útiles eran este tipo de soluciones era en la televisión.

En 1989 llevaron a cabo un proyecto que determinaría el rumbo de la empresa. Este fue sistema de automatización para mejorar la calidad de emisión y relevar a los operadores de la tensión de la operación manual. *Vector3* desarrolló e implementó un proyecto completo y a pesar que los ordenadores eran lentos (acababa de salir al mercado el procesador a 386Mhz), consiguió una precisión en las transiciones de +- 1 *frame* que hasta entonces solo había estado al alcance de los grandes operadores.

A partir de ahí han seguido evolucionando sus productos hasta el día de hoy.

### Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity

Software de continuidad **Vector Box DSX** [Vec1]. Es un sistema de continuidad ligero. Su monitorización sobre VGA (*Video Graphics Array*) a través de LAN le convierte en una pieza perfecta para mantener la continuidad en entornos distribuidos. Tiene la capacidad gráfica para poder añadir el canal faldones, relojes de informativos, resultados deportivos, *tickers* de noticias, meteorología, chat de sms y en general todos los gráficos que una televisión necesita en su continuidad.

Utiliza un *genlock* para la sincronización de fotogramas analógico. La entrada que utiliza es vídeo compuesto, Y/C ó YUV (con opción en digital) con dos entradas de audio XLR balanceados y a la salida dispone exactamente de lo mismo, vídeo compuesto, Y/C ó YUV (con opción en digital) con dos entradas de audio XLR balanceados. Para el almacenamiento de material utiliza disco SATA, pudiendose conectar hasta 4 de 250 GB.

**Vector Box DSX** dispone de una versión con capacidad para manejar vídeo en alta definición, pero el coste de esta mejora en el sistema es extremadamente alto lo que lo hace inviable en el presente proyecto [VecP3].

El software **Vector Box6000** [Vec2] que es la evolución de este software. También dispone de una versión para transmisión de vídeo en alta definición, pero su coste es desorbitado. El *genlock* que utiliza es digital ó analógico y como entrada tiene SDI digital con 8 pares de audio embebidos. Su salida es 2 SDI digital con 16 pares de audio embebidos. Permite hasta 8 discos Cheetah SCSI de 143 GB.

### 2.4.23 VSN

- Web: [www.vsn.es](http://www.vsn.es)
- Contacto: ¿? (¿?@email.com, teléfono¿?).

VSN es una empresa con sede en Barcelona y fue fundada en 1990 aunque originalmente lo hizo bajo el nombre de NRD. VSN es una empresa con larga experiencia en el sector de *broadcast*. De hecho, hoy en día más de 200 canales han digitalizado sus informativos con **VSNnews** y unos 400 canales de TV han automatizado su continuidad con **VSNmatic**.

VSN desarrolla sistemas abiertos y escalables, basados en tecnología abierta y no propietaria. El desarrollo de su primer modulo se dio en 1996 con **VSNmatic** y a partir de ahí hasta el día de hoy ha añadido numerosas funcionalidades para ayudar a la emisión de televisión.

## Soluciones aportadas por la empresa al esquema global:

- V/A playout and continuity

**VSNOneTv** [VsnP1]. Es una plataforma que agrupa las funcionalidades básicas para la emisión de televisión en una sola máquina. Entre estas funcionalidades destacan: control de ingesta y dispositivos de ingesta, *playout* y control de dispositivos de *playout*, generador de caracteres y gráficos. Es un sistema modular y todas estas funciones se añaden con sus módulos. A continuación se describen los módulos que componen **vsnOneTv** [Vsn1, Vsn2]:

**VSNmatic**. Es el layout diseñado para automatizar 24x7 la sala de continuidad. Tiene capacidad para reproducir vídeos multiformato desde disco duro y controlar dispositivos como matrices, VTR's, etc. Al controlar matrices de vídeo conectadas a su entrada, sus diferentes entradas puedan ser conmutadas. En principio, **VSN** recomienda la opción de usar matrices *Kramer*, por ser estas las más extendidas en el mercado y las más baratas.

**VSNmatic** se encarga de controlar todos los módulos anteriormente descritos mediante acciones de la escaleta. Esta escaleta se puede preparar incluso con meses de antelación con el modulo **VSNmatic remote**. Puede tener múltiples fuentes, combinar materiales provenientes del servidor **VSNmatic**, de líneas de exterior, VTR's, grabación automática de eventos (sería necesario además **VSNautorec**), publicidad automática y todo gestionado desde la escaleta de **VSNmatic**.

También recomiendan tener un sistema de archivo (**VSNarchive**) que se conectaría vía SCSI a la máquina (normalmente una caja para *rack*) donde estaría instalado el **VSNmatic**.

**VSNarchive**: Es el software de gestión documental y de archivo del sistema. Su avanzado *DAM (Digital Asset Manager)* cubre todos los procesos de ingesta, catalogación, almacenamiento *online*, *near line* y *offline*. Búsquedas avanzadas por *descriptores* y texto libre, compactados, prestamos de cintas, solicitudes y distribución. Incluye licencia servidor, licencia cliente (búsqueda y recuperación) y gestor de contenidos.

**VSNrecpro**: Es el software de ingesta al servidor **vsnmatic**. De entradas tiene Video SDI (1xBNC) y en Analógico: CVBS, Componentes YUV e Y/C. Como salida Video Digital SDI (1xBNC) y analógicas YUV, CVBS e Y/C.

**VSNcontrol**: Es un sistema para el control de dispositivos. Controla switchers, routers y VTR's para automatización 24x7. Controla hasta 5 VTRs via RS422 y un router-switcher via RS232. Incluye una tarjeta de 8 puertos RS422.

**VSNcg**: Es una tituladora automática o manual con efectos avanzados, fundidos y movimientos 2D, textos en *roll* y *crawl*, animaciones 3D e imágenes fijas TGA o BMP, con capas ilimitadas en tiempo real. Incluye

*plug-ins* con contadores para deportes, relojes, cuenta atrás etc. Su entrada de Video es SDI (1xBNC) y la salida SDI de video digital (1xBNC).

**VSNIg:** Software generador de logotipos que soporta logotipos en 2D y animaciones 3D. Su entrada de Video es SDI (1xBNC) y la salidas SDI de video digital(1xBNC).

La solución es escalable y la propia empresa es la encargada de la instalación e integración de todos sus productos. Una buena opción, sin duda, en caso de querer instalar un sistema profesional y de alta capacidad (es la opción que tomó en su día el ayuntamiento de Alcázar de San Juan).

## 3 Gestión de un canal de televisión con software libre

### 3.1 Objetivo

Ante la implantación de la televisión digital terrestre y la llegada del apagón analógico se plantea la posibilidad del nacimiento de pequeños canales de televisión de ámbito local. Ya bien sea en una pequeña ciudad o región, cabe la posibilidad de ofrecer contenidos de producción propia y gestionarlos por los propios miembros de las comunidades. Además, la sociedad de la información saca la creación y edición de *vídeos* del entorno profesional para convertirse en un hobby más. Ante esta situación se plantea una brecha tecnológica entre la creación de contenidos y su emisión al aire. Se puede recurrir a soluciones profesionales que permitan la emisión de los contenidos audiovisuales pero a un elevado coste económico. Ante esa brecha se ofrece una solución de bajo coste amparado por el uso de software libre.

El objetivo principal, es conseguir la emisión de contenidos “al aire” para su difusión de manera programada. Para ello se hará uso de dos programas de software libre instalados en un ordenador de gama media-alta, equipado adicionalmente con hardware diseñado para la transmisión de televisión digital. Uno de los programas se encargará de la automatización en la emisión de los contenidos mientras que el otro se ocupará propiamente de la adaptación de los contenidos para la emisión siguiendo el estándar de televisión digital terrestre *DVB-T*[Dvb1].

El software utilizado para una y otra tarea será *VideoLan Client (VLC)* [Vlc1] y *Opencaster* [Opc1] respectivamente. Debido a que *Opencaster* ha de ser instalado en un sistema operativo *Linux*, se deberá hacer uso de este sistema en el ordenador. Se necesitará, además, una tarjeta moduladora instalada en el ordenador, encargada de adaptar la señal para la transmisión y posterior recepción en los decodificadores de televisión digital terrestre.

El sistema funciona de la siguiente manera:

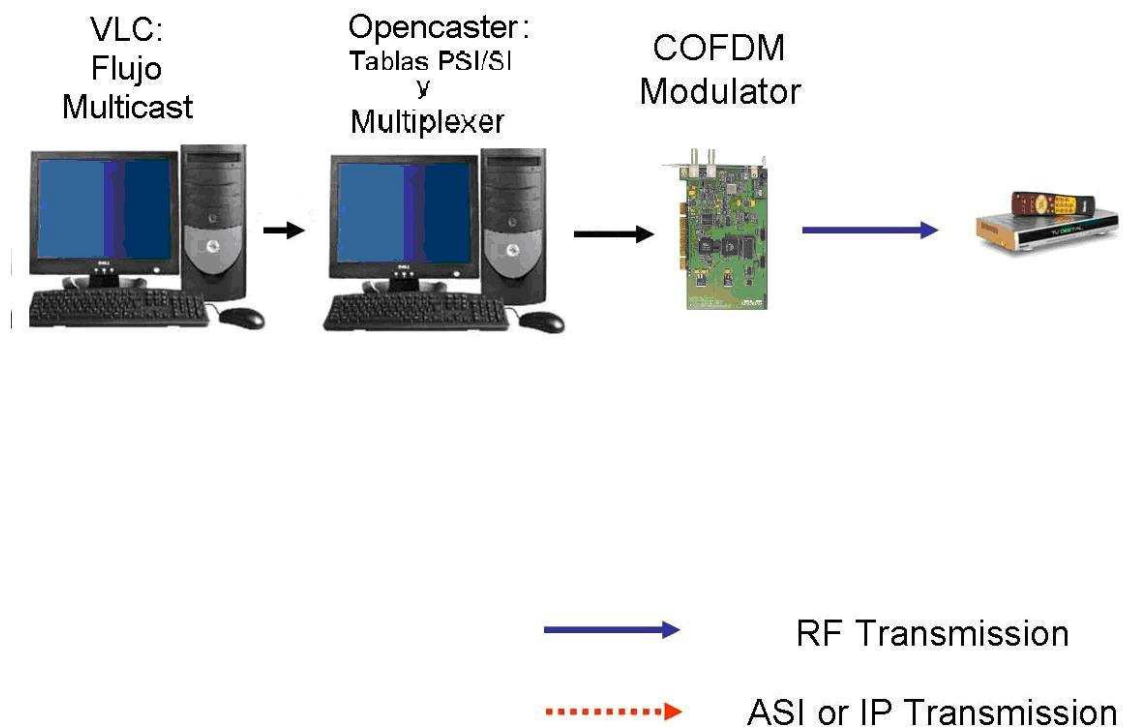
- El *VLC* sirve los *vídeos* de manera programada a través de una conexión multicast en red a *Opencaster*. Para ello *VLC* dispone de una interfaz web para la gestión y programación de la emisión de contenidos llamada *VideoLan Manager (VLM)*[Vlm1]. Gracias a la conexión multicast, el *VLC* y los contenidos pueden estar en un ordenador distinto al que correrá con *Opencaster*, liberando a este último de carga computacional, pero ambos deberán estar en la misma red de área local. Aunque también existe la posibilidad de correr *VLC* y *Opencaster* en el mismo equipo, a costa de una gran carga computacional.
- *Opencaster* se ocupa de realizar la adaptación al estándar de emisión de televisión digital terrestre *DVB-T (Digital Video Broadcasting* –



*Terrestrial*). Además, *Opencaster* permite incluir información sobre los contenidos emitidos (*Electronic Program Guide EPG*) [Epg1] y aplicaciones multimedia.

- Por ultimo la tarjeta moduladora se encarga de la adaptación de la señal a emitir a la modulación usada por el estándar de televisión digital terrestre para su difusión *COFDM* (*Coded Orthogonal Frequency Division Multiplexing*) [Cof1].

Se puede observar en la *figura 6* el esquema del sistema.



**Figura 6. Esquema del sistema de continuidad**

## 4 Gestión de la escaleta mediante el VLC

El objetivo es realizar la automatización de la emisión de un canal de televisión digital por medio de software de libre distribución. Para ello, se aprovecha las utilidades del reproductor *VLC media player* (a partir de ahora lo llamaremos *VLC*) que es un reproductor multimedia del proyecto *VídeoLAN*, distribuido bajo licencia GPL (*General Public License*). Soporta muchos *códecs* de audio y vídeo, así como diferentes tipos de archivos, además de DVD, VCD y varios protocolos *streaming*. Estos protocolos de *streaming* se podrán utilizar para que el ordenador funcione como servidor *unicast* o *multicast*.

Además de las funciones para hacer el *streaming* dentro del *VLC*, se dispone también de otra funcionalidad llamada *VídeoLAN Manager (VLM)*, que sirve para realizar la automatización de la emisión de vídeo. Con él, se podrá marcar la hora comienzo de emisión, introducción de nuevos vídeo y en general las tareas básicas para poder hacer la automatización del canal. Esta herramienta es bastante limitada, ya que no es un software creado para esta tarea en concreto, pero sus características son suficientes para crear una escaleta sencilla.

Para recoger el *streaming* que se manda a través del *VLC*, se utiliza un conjunto de herramientas de libre distribución enfocadas a la emisión de contenidos multimedia. Estas herramientas se utilizan para procesar, multiplexar y emitir un flujo de datos *MPEG-2 Transport stream (MPEG-2 TS)*. Este conjunto de herramientas se llama *Opencaster*.

Se usán las herramientas de *Opencaster* para recoger el vídeo, añadir todas las tablas *PSI-SI* y multiplexarlo con una aplicación interactiva.

El *Opencaster* solo puede recoger el vídeo desde una dirección *multicast*, así que el vídeo que el *VLC* lanza por *streaming* tendrá que ir a una dirección *multicast*. En el caso de estudio se lanza el vídeo a la dirección 224.0.0.1.

Una vez que *Opencaster* hace todas las adaptaciones necesarias, lanza el *streaming* a la tarjeta moduladora de *dektec* que está instalada en el ordenador, y de ahí llega directamente al receptor Televisión Digital Terrestre (TDT).

En la *Figura 6* se puede ver el esquema que se quiere realizar.

## 4.1 VLC

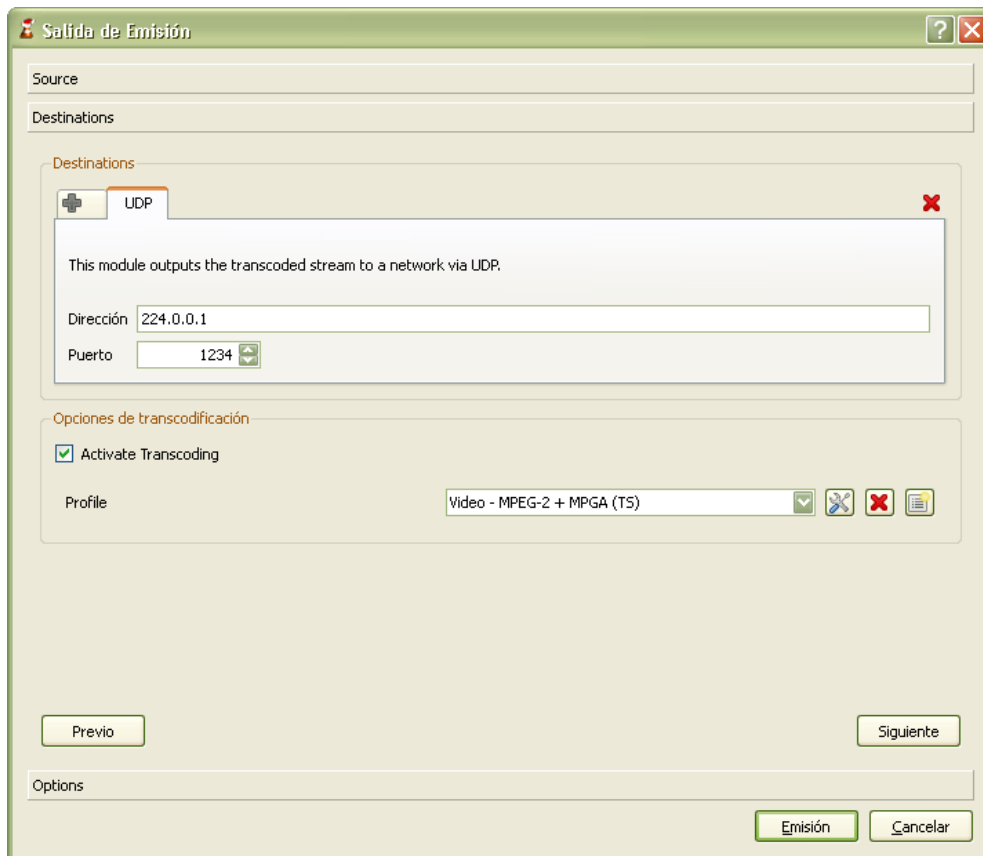
El VLC ofrece una solución de software libre completa para transmisión de vídeo. Puede ser usado como servidor para transmitir *MPEG-1*, *MPEG-2* y *MPEG-4*, *DVD* y vídeo en vivo sobre la red en *unicast* o *multicast*, o utilizarse como un cliente para recibir y decodificar *streams* de vídeos en cualquier sistema operativo.

La forma para hacer un *stream* de vídeo es muy sencilla con el VLC: primero hay que seleccionar la opción de emisión como se muestra en la *Figura 7* y aparecerá una ventana para poder seleccionar el vídeo a transmitir.



**Figura 7**

A continuación como se puede ver en la *Figura 8*, se podrá elegir el destino, el tipo de protocolo que se utiliza para la emisión y el tipo de transcodificación. Para el caso de estudio se elige el protocolo *UDP* y se lanza a una dirección *multicast*, ya que el *Opencaster* solo puede recoger el *stream* desde una dirección *multicast*. También hay que elegir un puerto para la transmisión.



**Figura 8**

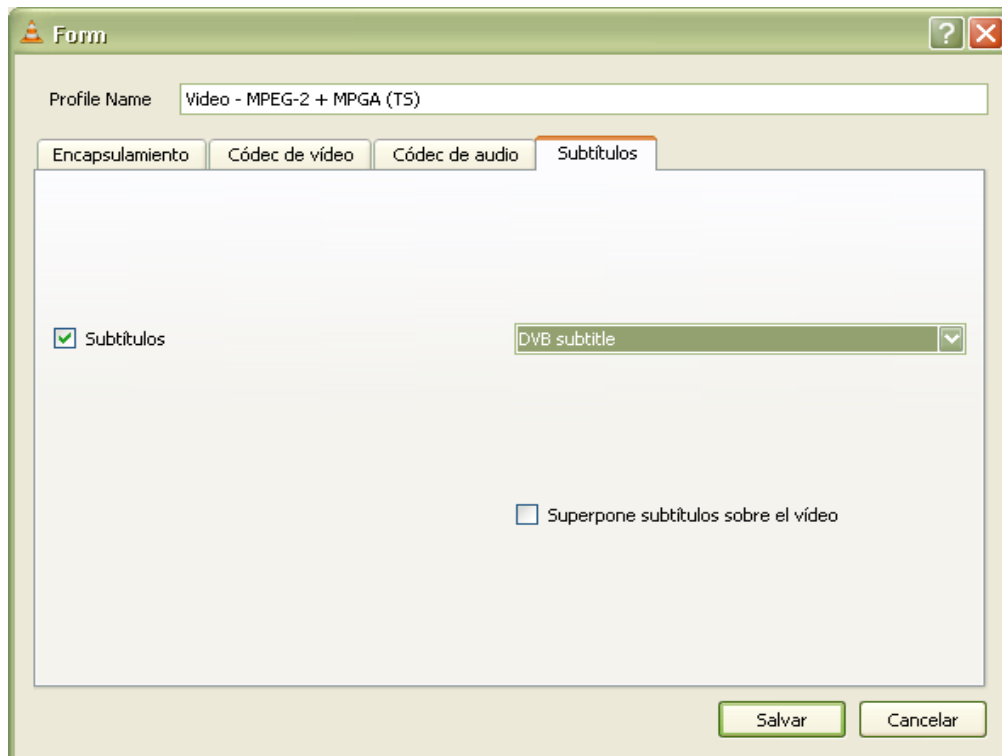
Ahora hay que elegir las opciones de transcodificación en la misma ventana. Como es una transmisión de televisión digital, se elige *MPEG-2 TS*, y se selecciona *propiedades* para ajustar los valores de la transcodificación, como: *bitrate*, *codec* de vídeo y audio, alto y ancho del vídeo etc. En las Figuras 9, 10 y 11 se observan estas opciones:

The screenshot shows the 'Form' dialog box in VLC media player. The 'Profile Name' field is set to 'Video - MPEG-2 + MPGA (TS)'. The 'Encapsulamiento' tab is selected, and the 'Código de vídeo' sub-tab is active. The 'Vídeo' checkbox is checked, and the 'Keep original video track' checkbox is unchecked. The 'Código' dropdown is set to 'MPEG-2'. The 'Tasa de Bits' (Bitrate) is set to '5000 kb/s' and the 'Frame Rate' is set to '25,00 fps'. The 'Resolución' (Resolution) section contains a message: 'You just need to fill one of the three following parameters, VLC will autodetect the other using the original aspect ratio'. Below this, the 'Escala' (Scale) dropdown is set to '1', and the 'Anchura' (Width) and 'Altura' (Height) fields are both set to '0'. At the bottom right, there are 'Salvar' (Save) and 'Cancelar' (Cancel) buttons.

**Figura 9**

The screenshot shows the 'Form' dialog box in VLC media player, with the 'Código de audio' sub-tab active. The 'Audio' checkbox is checked, and the 'Keep original audio track' checkbox is unchecked. The 'Código' dropdown is set to 'MPEG Audio'. The 'Tasa de Bits' (Bitrate) is set to '128 kb/s', the 'Canales' (Channels) is set to '2', and the 'Sample Rate' is set to '44100'. At the bottom right, there are 'Salvar' (Save) and 'Cancelar' (Cancel) buttons.

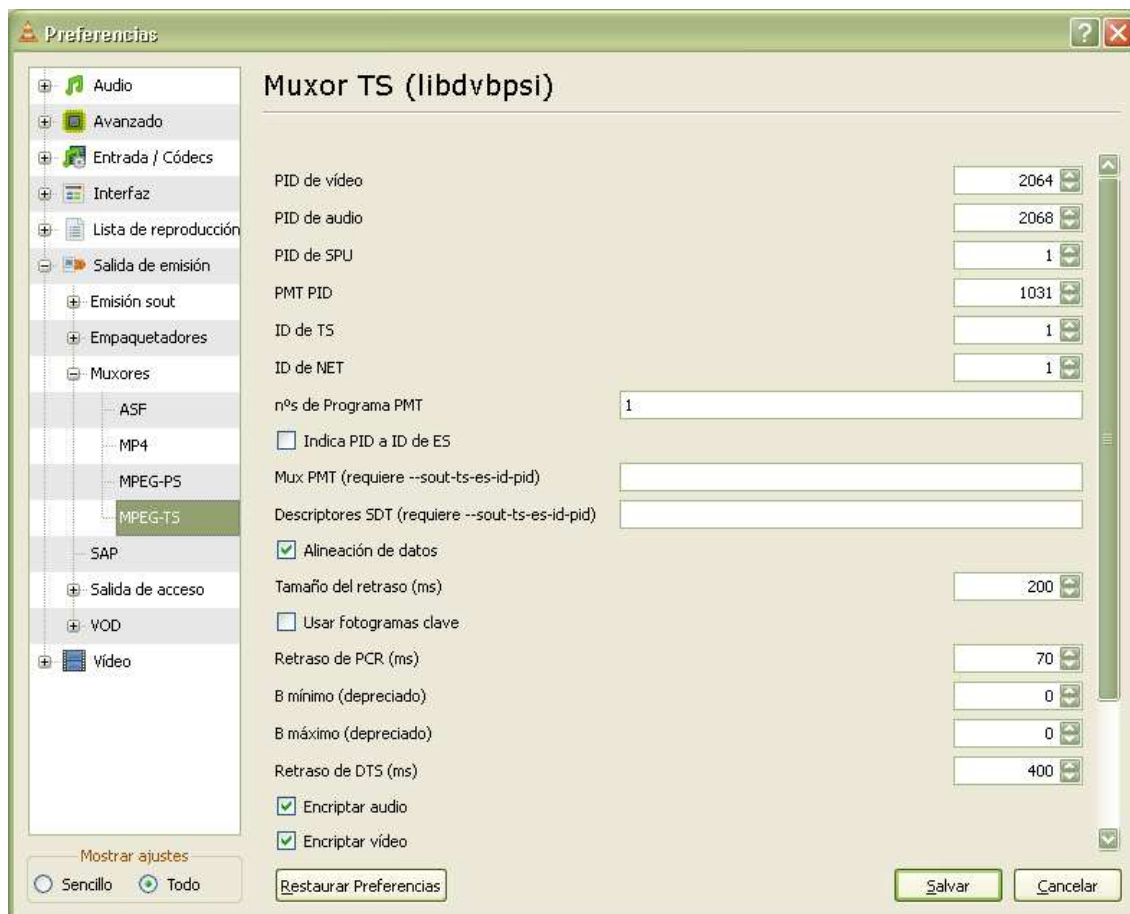
**Figura 10**



**Figura 11**

En un principio aparecen los valores por defecto pero estos podrán ser ajustados. Se podrá ajustar las tasas de *bitrate* para el audio y el vídeo por separado. En la recepción con el *Opencaster* habrá que tener en cuenta que el *stream* que llega tendrá un *bitrate* que será la suma de los *bitrates* de audio y vídeo. Es muy importante tener en cuenta el *bitrate* del *stream* que se lanza para luego poder hacer una correcta recogida con el *Opencaster*. Si el valor que se pone en el *Opencaster* no es el correcto, el vídeo se verá con cierta pixelación y con una descompensación entre el vídeo y el audio.

Además, hay que definir las propiedades del *MPEG-2 TS*. Esto se debe hacer previamente, en las preferencias avanzadas del *VLC*. En el caso de estudio se ha elegido la *PID* de vídeo de 2064 y la *PID* de audio de 2068. Estos valores son los únicos que nos interesan ya luego son las 2 *PID* que se van a filtrar con el *Opencaster*. En la ventana que aparece en la *Figura 12*, se puede observar como definir estas propiedades en el *VLC*:



**Figura 12**

Una vez ajustados los parámetros, ya se puede presionar sobre el botón *emisión* y se podrá recoger el *stream* en cualquier ordenador de la red *multicast* ya que, como se ha comentado, se está lanzando a una dirección *multicast*. El propio VLC tiene una herramienta de captura que se puede utilizar, pero para nuestro objetivo, que es hacer una emisión de televisión digital, utilizamos el *Opencaster* para recoger el *stream*, ya que éste nos hará la adaptación de la señal.

Además, el VLC tiene la aplicación *VLM* que permite hacer la automatización de la emisión.



## 4.2 VLM

### 4.2.1 Introducción comandos VLM

Para poder controlar el *VLC* de manera automatizada y manejar varios flujos de *streaming* (los contenidos que se van a emitir) con solamente un *VLC* abierto tenemos la posibilidad de usar el *VLM*.

El *VLM* es un conjunto de instrucciones que son capaces de administrar los contenidos a emitir por el *VLC*. Para poder transmitir estas instrucciones al *VLC* se usan 2 tipos de interfaces:

- El interfaz *Telnet*, con el que se puede mandar los mensajes abriendo una comunicación *Telnet* con el *VLC*.
- El interfaz *http* o Interfaz web, con el que se puede mandar los mensajes a través de una pagina web que permite mandar los mensajes mediante comunicación *http*.

En primer lugar se explicará la estructura de las instrucciones *VLM* y el significado de las instrucciones, dando ejemplos de archivos de configuración *VLM* con las distintas posibilidades que da.

Una vez explicadas todas las instrucciones de *VLM*, se presentará, solamente el manejo de la interfaz web, ya que es más fácil de utilizar a la hora de crear los archivos de configuración *VLM* y automatizar el proceso.

### 4.2.2 ¿Que es un archivo de configuración VLM?

Un archivo de configuración *VLM* es un archivo de texto donde están escritos los comandos que se mandarán al *VLC* para que sea manejado. Al modificar este archivo de configuración se logrará que el *VLC* haga lo que se pretende de forma automatizada, bien sea reproducir, pausar ó parar un vídeo ó un conjunto de vídeos.

#### • Estructura de un archivo de configuración VLM

Los archivos *VLM* tienen 2 tipos de elementos:

- Elementos de medios de emisión o **elementos de *broadcast***, que definirán las propiedades de los vídeos que queremos emitir.
- **Elementos de programación**, que serán las fechas y horas en los que se ejecutará un cierto comando de control.

Además, el *VLM* define los **comandos de control** que actúan sobre los elementos de *broadcast*. Estos comandos se pueden introducir en los elementos de programación para que actúen sobre los elementos de *broadcast* en una fecha y hora determinadas.

#### • Elementos de *broadcast*

Para los elementos de *broadcast* se deberá definir la **entrada** y la **salida**.

- La entrada será el vídeo a emitir. También existe la posibilidad de que la entrada sean varios vídeos. Si la entrada son varios vídeos, estos se emitirán de manera consecutiva en el orden en que se introduzcan.
- La salida serán los parámetros de salida que usara el VLC a la hora de emitir. Tipo de *streaming* y recodificación si es necesaria.
- La salida, en el caso de estudio, será la misma para todos los elementos de *broadcast* ya que nuestro propósito es emitir un canal de televisión digital. Por esta razón, la salida ha de ser la misma para todos los contenidos que queremos emitir. Los parámetros que se eligen dependen, además, de las necesidades que marca el *Opencaster* para inyectar el canal al aire.

#### ○ Ejemplo 1

Este es un ejemplo básico de cómo crear un elemento de *broadcast*.

```
new ejemplo broadcast enabled
setup ejemplo input "D:\Vídeos\vídeo.avi"
setup ejemplo output
#transcode{vcodec=mp2v,vb=4096,scale=1,acodec=mpga,ab=128,channels=2}:
std{access=udp,mux=ts,dst=224.0.0.1:1234}
```

En el *ejemplo 1* se crea un elemento de *broadcast* con el nombre *ejemplo* haciendo uso del comando *VLN new*. Para poder usar el elemento se debe habilitar, lo que se consigue agregándole la propiedad **enabled**.

Al elemento de *broadcast ejemplo* hay que definirle la entrada que va a usar con la inserción **setup ejemplo input** "D:\Vídeos\vídeo.avi". La salida se define de manera similar con **setup ejemplo output...**

La salida, en este caso, recodifica el vídeo a *MPEG-2 TS*, que es el codec para emisiones de televisión digital, con el comando **transcode{...}**. También hay que definir las opciones de salida del *streaming*. El *streaming* que se va a usar, es a través del protocolo de comunicación *UDP* con destino una dirección *multicast*. Esto es así porque se tendrá escuchando en esa dirección *multicast* una aplicación que será la encargada de inyectarlo para su emisión "al aire".

A la hora de definir el elemento de *broadcast*, se tiene la posibilidad de que haga una emisión en "lazo", es decir, que cuando acabe el *stream* vuelva de nuevo a empezar una y otra vez. Esto se logra dotándole de la propiedad **loop**.

#### ○ Ejemplo 2

```
new ejemplo broadcast enabled loop
setup ejemplo input "D:\vídeos\vídeo1.avi"
setup ejemplo input "D:\vídeos\vídeo2.avi"
```

```
setup ejemplo output  
#transcode{vcodec=mp2v,vb=4096,scale=1,acodec=mpga,ab=128,channels=2}:  
#std{access=udp,mux=ts,dst=224.0.0.1:1234}
```

Ahora en el *ejemplo 2* se tienen dos vídeos dentro de un elemento de *broadcast*. Estos se reproducirán de manera consecutiva, primero *vídeo 1* y luego *vídeo 2*. Además, como el elemento de *broadcast* tiene la propiedad **loop**, los vídeos se emitirán en lazo.

- **Elementos de programación**

A los elementos de programación se les debe dar un nombre también y definirlos proporcionando la fecha y hora a la que se quiere que se emitan. Además, se puede establecer la repetición de la programación asignándole el tiempo que ha de pasar para que el elemento se repita y el número veces que se quiere que se repita (por ejemplo, para repetir un elemento de programación todos los días se debe poner 1 día en el tiempo de repetición y repeticiones infinitas).

Una vez definidas, hay que establecer los comandos de control que actuarán a la hora programada especificando sobre qué elemento de *broadcast* han de actuar.

Una misma programación puede actuar sobre distintos elementos de *broadcast* y con distintos comandos de control. Por ejemplo, se puede parar un elemento de *broadcast* y comenzar la emisión de otro con el mismo elemento de programación, es decir, en el mismo momento.

- *Ejemplo 3*

```
new ejemplo broadcast enabled  
setup ejemplo input "D:\vídeos\vídeo1.avi"  
setup ejemplo input "D:\vídeos\vídeo2.avi"  
setup ejemplo output  
#transcode{vcodec=mp2v,vb=4096,scale=1,acodec=mpga,ab=128,channels=2}:  
std{access=udp,mux=ts,dst=224.0.0.1:1234}
```

```
new ejemplo2 broadcast enabled  
setup ejemplo2 input "D:\vídeos\vídeo3.avi"  
setup ejemplo2 output  
#transcode{vcodec=mp2v,vb=4096,scale=1,acodec=mpga,ab=128,channels=2}:  
std{access=udp,mux=ts,dst=224.0.0.1:1234}
```

```
new empezar schedule date 2009/12/14-14:0:0 enabled  
setup empezar period 0/0/1-0:0:0
```

```
setup empezar append control ejemplo play
```

```
new parar schedule date 2009/12/14-15:0:0 enabled  
setup parar period 0/0/1-0:0:0  
setup parar repeat 0
```

```
setup parar append control ejemplo2 play
```

```
setup parar append control ejemplo stop
```

En este ejemplo se tienen dos elementos de *broadcast ejemplo* y *ejemplo2*.

Luego se crea un elemento de programación llamado *empezar* con fecha 14 de diciembre de 2009 a las 14 horas con un periodo de repetición de 1 día y que se repetirá sin fin. También se crea otro elemento de programación llamado *parar* el mismo día una hora mas tarde. Este elemento de programación tiene una repetición de un día pero solo se lanzará una vez (la primera que se lance). Si se quiere que se repita solo al día siguiente tendríamos que poner *repeat 1*.

En último lugar, se definen los comandos de control que actúan sobre los elementos de *broadcast ejemplo* y *ejemplo2*.

En el ejemplo, se ha configurado que el elemento de programación *empezar* lance el elemento de *broadcast ejemplo*. Para el elemento de programación *parar* se ha configurado que pare el elemento de *broadcast ejemplo* y que, a la vez, lance el elemento de *broadcast ejemplo2*. Esta es la manera en que se concatenan los elementos de *broadcast* ya que así no existe ningún lapso de tiempo entre los dos.

El resultado es que a la hora especificada por *empezar* comienza la emisión de *ejemplo* y una hora después se para y comienza la emisión de *ejemplo2*.

Existe un pequeño problema a la hora de hacer la concatenación de los elementos de *broadcast* programada, y es que es difícil de encontrar el punto exacto donde se quiere parar el vídeo, ya que este software no da herramientas para ello.

Además, si el primer elemento de *broadcast* tuviera una duración total menor que el tiempo que pasa entre el comienzo del primer elemento de programación y el siguiente, existirá un hueco donde no se emitiría nada. En el ejemplo, si el elemento de *broadcast ejemplo* dura menos de una hora el tiempo restante hasta que pase la hora no habrá nada en emisión.

Es decir, si se quiere que los vídeos se emitan enteros y sin huecos se tienen dos posibilidades, ó bien introducirlos en el mismo elemento de *broadcast*, ó bien calcular la duración de los vídeos y programar la concatenación de forma exacta.

Los comandos *VLM* también dan la posibilidad de enviar un comando de control sobre un elemento de *broadcast* en cualquier momento que se quiera, sin tener que crear un elemento de programación. Para ello se tiene simplemente que mandar el comando adecuado al *VLC*.

```
control ejemplo play  
control ejemplo pause  
control ejemplo play  
control ejemplo play seek 17  
control ejemplo stop
```

Cuando se ejecute el primer comando, comenzará la emisión de *ejemplo*.

Cuando se ejecute el segundo comando, pausará la emisión de *ejemplo*.

Cuando se ejecute el tercer comando, continuará la emisión de *ejemplo* en el punto donde fue pausado.

Cuando se ejecute el tercer comando, la emisión saltará al momento en el que lleva un 17% de la entrada de la lista que se estuviera reproduciendo en ese momento y seguirá reproduciéndose.

Cuando se ejecute el tercer comando, parará la emisión de *ejemplo*.

También hay que tener cuidado de no emitir dos medios a la vez por la misma salida, ya que estos colisionarían y no se vería ninguno de los dos.

### 4.2.3 Comandos VLM

Una vez que se ha visto de manera general como funciona los archivos *VLM*, se verá específicamente los comandos existentes.

Se comienzan con los comandos básicos:

- **new (nombre) vod|broadcast|schedule [propiedades]:** Para crear un elemento nuevo. Puede ser vídeo bajo demanda (*vod*), un elemento de *broadcast* o una programación *schedule*. Se pueden especificar propiedades en esta línea o mas adelante usando el comando **setup**.
- **setup (nombre) (propiedades)** Establece las propiedades del elemento:
  - **del (nombre) |all|media|schedule:** Borra un elemento o un grupo de elementos. Si el elemento no esta parado lo para antes de ser borrado.
    - **del (nombre):** Borra el elemento con el nombre que se indique.
    - **del all:** Borra todos los elementos.
    - **del media:** Borra todos los elementos de *broadcast*.
    - **del schedule:** Borra todos los elementos de programación.
- **control (nombre) (comando):** Cambia el estado del elemento de programación que se le indique.

Ahora se verá específicamente algunas propiedades y los comandos de control.

Para los elementos de *broadcast*:

- **input (nombre)**: Añade una entrada al final de la lista de vídeos, con ello se pueden concatenar vídeos uno detrás de otro.
- **output (nombre)**: Define la salida con el nombre que se le indique. La sintaxis es la propia del VLC ":sout=..." pero no hay que poner ":sout=".
- **enabled|disabled**: Habilita o deshabilita el elemento de *broadcast*. Si se deshabilita no se puede emitir ni hacer nada con ella.
- **loop|unloop (solo para broadcast)**: Si un elemento de *broadcast* con la opción loop recibe el comando de **play**, automáticamente volverá a comenzar el *play* de la lista de entradas una vez llegue al final.

Para los elementos de Programación:

- **enabled|disabled**: Habilita ó deshabilita un elemento programación. Mientras no este habilitado ninguno de los demás comandos tendrá efectos sobre este elemento de programación.
- **append(command\_until\_rest\_of\_the\_line)**: Añade un comando.
- **date(año)/(mes)/(día)-(hora):(minutos):(segundos)|now**: Especifica la primera fecha en el que la programación será lanzada. Ha de hacerse con el siguiente formato **(año)/(mes)/(día)-(hora):(minutos):(segundos)** (ejemplo 2004/11/16-00:43:12) o usando la palabra clave **now**. Si se usa **now** la programación será lanzada tan pronto como se pueda (por ejemplo, tan pronto como sea habilitada) y la fecha en ese momento será usada como la primera fecha de la programación.
- **period (años = 12 meses)/(meses = 30 días)/(días)-(horas):(minutos):(segundos)**: Especifica el periodo de tiempo que una programación ha de esperar para volver a lanzarse otra vez (los meses se consideran como 30 días y los años como 12 meses). Si el periodo se especifica sin la propiedad de **repeat**, la programación se repetirá sin fin con el periodo especificado.
- **repeat (numero de repeticiones)**: Especifica el número de veces que la programación volverá a ser lanzada. Por ejemplo, si una programación tiene un **repeat** de 11 ésta será lanzada 12 veces, la inicial mas las 11 repeticiones.

Los comandos de control son:

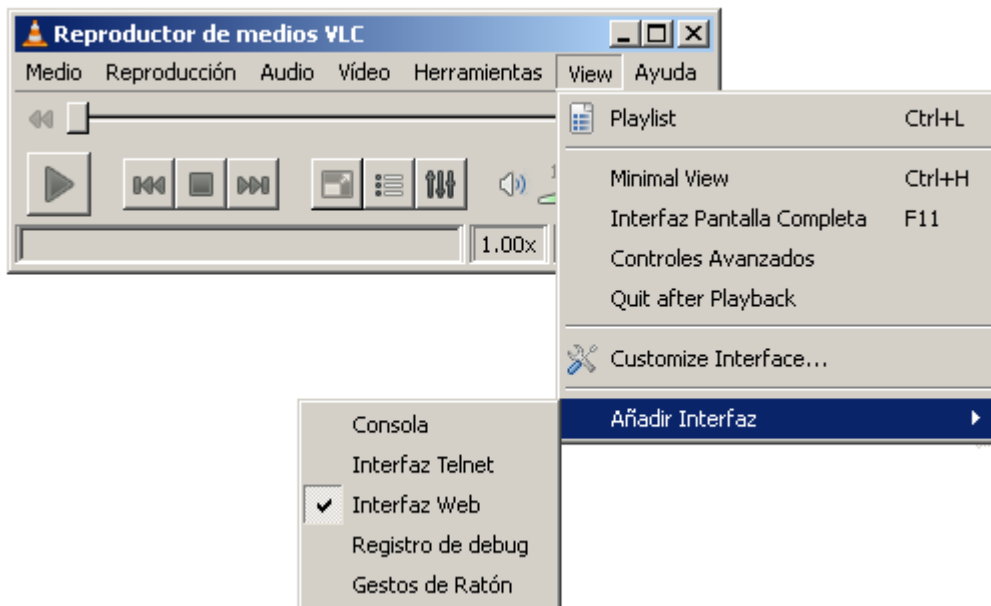
- **play**: Comienzo de la emisión del elemento de *broadcast*. Comienza con la primera entrada de la lista de entradas y sigue con la segunda, etc.
- **pause**: Pone a la emisión en pausa.
- **stop**: Para la emisión.
- **Seek**: Hace que la emisión comience en un punto determinado del vídeo. Este punto se indicara con por medio de un porcentaje, por ejemplo si queremos que empiece a mitad del vídeo se deberá poner **seek 50**

## 4.2.4 Uso del Interfaz Web

Ahora se verá como hacer archivos de configuración *VLM* y mandar comandos de control con el uso de la interfaz web del *VLC*.

### 4.2.4.1 Activar el interfaz Web

Primero se tiene que activar el interfaz web del *VLC*, como se describe en la *Figura 13*:



**Figura 13**

Primero se tiene que activar el interfaz web del *VLC*.

### 4.2.4.2 Descripción General

Una vez activada la interfaz web se debe abrir un navegador de Internet y poner como dirección <http://localhost:8080/vlm.html> ó <http://127.0.0.1:8080/vlm.html>, 8080 es el puerto a través del cual se accede al interfaz. Este puerto puede ser cambiado en las preferencias del *VLC*. Esto sería para hacer la conexión mediante *localhost*, pero también se puede hacer vía Internet. En este caso sería añadiendo un interfaz *Telnet*, que también se puede ver en la *Figura 13* como se hace.

Una vez hecho esto se tendrá una página web en la que se puede ver varias partes. Una primera parte donde se definirán los comandos a mandar al *VLC* (*VLM Interface*) y con el que se podrá crear distintos elementos de *broadcast* (***Broadcast Elements***) y elementos de programación (***Schedule Elements***). También se podrá crear elementos para vídeo bajo demanda pero ese no es nuestro propósito (*Vídeo on Demand Elements*).

A continuación se puede observar en la *Figura 14* qué aspecto tiene el interfaz web del *VLM*:



**VLC media player - VLM interface**

VLM command:

Broadcast Video on Demand Schedule Other Hide VLM helper

New broadcast element

Name  ☐ Enable ☐ Loop

Input

Output

**Broadcast Elements** ↺

**Video on Demand Elements** ↺

**Schedule Elements** ↺

**Figura 14**

Pinchando en **Broadcast** se crea un elemento de *broadcast* en cambio si se pincha en **Vídeo on Demand** o **Schedule** se podrá crear un elemento de vídeo bajo demanda o un elementos de programación, dándoles a todos ellos las propiedades deseadas. Pinchando en **Other** se podrá añadir comandos *VLM* escritos previamente y mandarlos al *VLC*. Además, se podrá ver cómo quedara el archivo de configuración *VLM* si se pincha en **Export VLM commands**. Esto se puede ver las 2 Figuras siguientes:

VLC media player - VLM interface

VLM command:

Broadcast Video on Demand **Schedule** Other Hide VLM helper

New schedule

Name  ☐ Enable

Schedule date: ☐ Now

Year:  Month:  Day:  Hour:  Minute:  Second:

Schedule period:

☐ Repeat

Year:  Month:  Day:  Hour:  Minute:  Second:

Number of repetitions (use 0 to repeat endlessly):

**Broadcast Elements** ↻

**Video on Demand Elements** ↻

Figura 15

VLC media player - VLM interface

VLM command:

Broadcast Video on Demand Schedule Other Hide VLM helper

#paste your VLM commands here

Figura 16

#### 4.2.4.3 Crear un elemento de *broadcast*

Veamos como crear un elemento nuevo de *broadcast* y modificar sus propiedades. En este ejemplo se crea el elemento de *broadcast* y se le asocia un vídeo (entrada *Input*). Luego se le asigna otro vídeo que se reproducirá a continuación del primero, como si fuera una lista. Los dos vídeos forman parte así del mismo elemento de *broadcast*.

Para empezar, se empieza pinchando en **Broadcast** en la zona principal del *VLM interface* y con ello aparece **New broadcast Element** (Figura 14). Ahora hay que darle un nombre. A continuación, se pincha en **Enable**, para que esté habilitado y en **Loop**, si lo que se desea es que se emita en lazo. Se observa que al darle el nombre e ir completando las opciones, estas van apareciendo en el recuadro **VLM command**.

Una vez completado esto se debe definir al menos una entrada (el vídeo que se quiera reproducir), para ello se pincha en **Edit**. Al pinchar en **Edit** aparecerá la Figura 17, en la que aparecen encima de *VLM Interface* las opciones para definir una entrada *Input*. La entrada puede ser un archivo, un disco (CD o DVD) o bien una dirección de red en la que se esta recibiendo un canal de *streaming*.

En este caso, interesa que sea un archivo almacenado en el disco duro de nuestro ordenador, así que se pincha en **File** y después en **Browse** para elegir la ubicación del archivo (Figura 17 Pasos 1 y 2). La función de **Browse** es muy rudimentaria, así que hay que poner mucha atención cuando se esta utilizando. Es difícil diferenciar entre archivos y carpetas (las carpetas tienen al lado del nombre la palabra *select* entre paréntesis) y además, no tiene funciones como *ir a la carpeta anterior* aunque si dispone

de *ir a la carpeta superior*, lo cual hace un poco incomoda la navegación por los archivos.

**Input**

Input (MRL)

File Disc Network Fixed image

Open File

File name

**VLC media player - VLM interface**

VLM command:

Broadcast Video on Demand Schedule Other Hide VLM helper

New broadcast element

Name  ☒ Enable ☐ Loop

Input

Output

**Figura 17**

Una vez elegido el archivo de vídeo, se puede presionar **ok** (Figura 17 Paso 3) y con esto ya se tiene definida la entrada (Figura 18). Se observa de nuevo cómo se completa el *VLM Command* del elemento con la entrada definida.

**VLC media player - VLM interface**

VLM command:

Broadcast Video on Demand Schedule Other Hide VLM helper

New broadcast element

Name  ☒ Enable ☐ Loop

Input

Output

**Broadcast Elements**

**Video on Demand Elements**

**Schedule Elements**

**Figura 18**

Una vez realizado esto hay que definir la salida. En este ejemplo, se hará lo mismo que en los ejemplos anteriores de los archivos de configuración *VLM*, es decir, sacar la señal a una dirección de red en *streaming* (en este caso *multicast*) y además se re-codificar a formato *MPEG-2 TS* con un *bitrate* dado, para el vídeo y audio. Todo esto se observa en la *Figura 19*. Para hacer una prueba se puede visualizar la señal en el propio ordenador pinchando en **Display**, en vez de en *UDP 224.0.0.1 puerto 1234*, así saca la salida por el display del *VLC*.

**Stream Output**

Destination (MRL) `:sout=#transcode{vcodec=mp2v,vb=4096,scale=1,acodec=n`

Ok Cancel Reset Hide sout interface

Stream Output Helper

☐ Display

☐ File File name

☐ HTTP Address  Port

☐ MMSH Address  Port

☐ RTP Address  Port

☒ UDP Address  Port

☐ Default ☒ MPEG TS ☐ MPEG PS ☐ MPEG 1 ☐ OGG

☐ ASF ☐ MP4 ☐ MOV ☐ WAV ☐ Raw

☒ Video Codec  Bitrate (kb/s)  Scale

☒ Audio Codec  Bitrate (kb/s)  Channels

☐ Subtitles Codec  ☐ Subtitles overlay

☐ SAP announce

Group name  Channel name

☐ Select all elementary streams

Time-To-Live (TTL)

**Figura 19**

Una vez que está definida la salida (**output**) se presiona en **ok** en la parte superior para que escriba el comando *VLM* y quede definido el *output* para el elemento de *broadcast*.

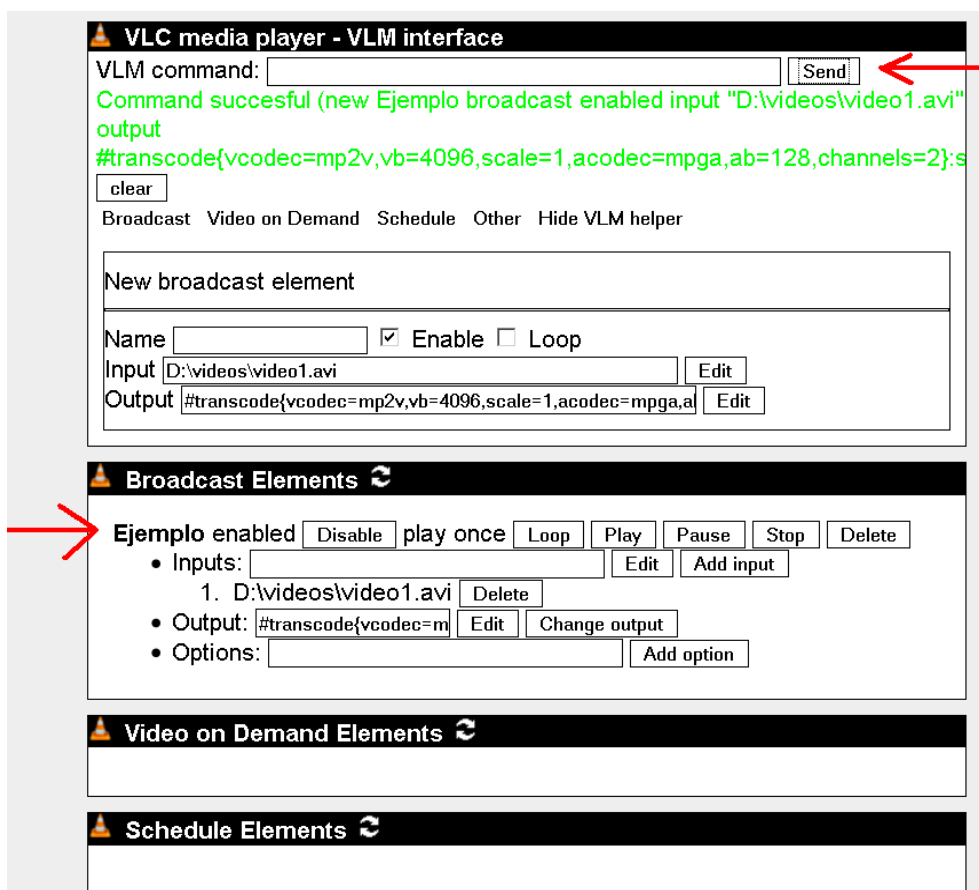
Ahora ya está definido por completo el elemento *broadcast*, así que se puede pinchar en **Send** para mandar el *VLM Command* al *VLC* y de esta manera el elemento se crea. Con todo esto queda listo para poder ser emitido cuando se desee (*Figura 20*).

Si todo está bien definido aparecerá un mensaje en verde y el elemento de *broadcast* creado con nombre *ejemplo* aparecerá en la parte de los **Broadcast elements**.

En este ejemplo se ha elegido un vídeo que necesita transcodificación pero podría ser que se quisiera enviar un vídeo que ya estuviera codificado en *MPEG-2 TS* con lo cual no haría falta codificarlo con el *VLC*. Para este caso

lo que habría que hacer es deseleccionar las opciones de **Vídeo Codec** y de **Audio Codec** y entonces se envía el vídeo sin codificar.

Una vez creado el elemento de *broadcast*, este puede ser modificado en la interfaz web. De esta manera aparecen botones que permiten: habilitarlo/deshabilitarlo, emitirlo una vez/ponerlo en lazo, comenzar emisión, pausarla, detenerla, borrar el elemento de *broadcast*, añadir *inputs* y añadir opciones como se puede ver en la *Figura 20*:



**Figura 20**

Ahora se pasará a asignar un segundo vídeo al elemento de *broadcast*. Para añadir otra entrada al elemento de *broadcast* es importante seguir los pasos, tal y como se muestra en la *Figura 21*, sino el programa dará funcionamientos no deseados. Primero se debe pinchar **Edit** en el elemento para elegir de nuevo una entrada. Se Tiene que seguir los mismos pasos que se han definido para añadir el primer vídeo del elemento de *broadcast*, es decir de nuevo pinchar en **file** y pinchando en **Browse** elegir el archivo, una vez elegido **Ok** con lo que aparecerá el elemento en la parte de *Inputs* del elemento de *broadcast*. Todavía no está añadido, el último paso será pinchar en **Add input** para que se añada a la lista de entradas (*Figura 22*).

En la *Figura 23* se observa como aparecen las dos entradas una detrás de otra como una lista.

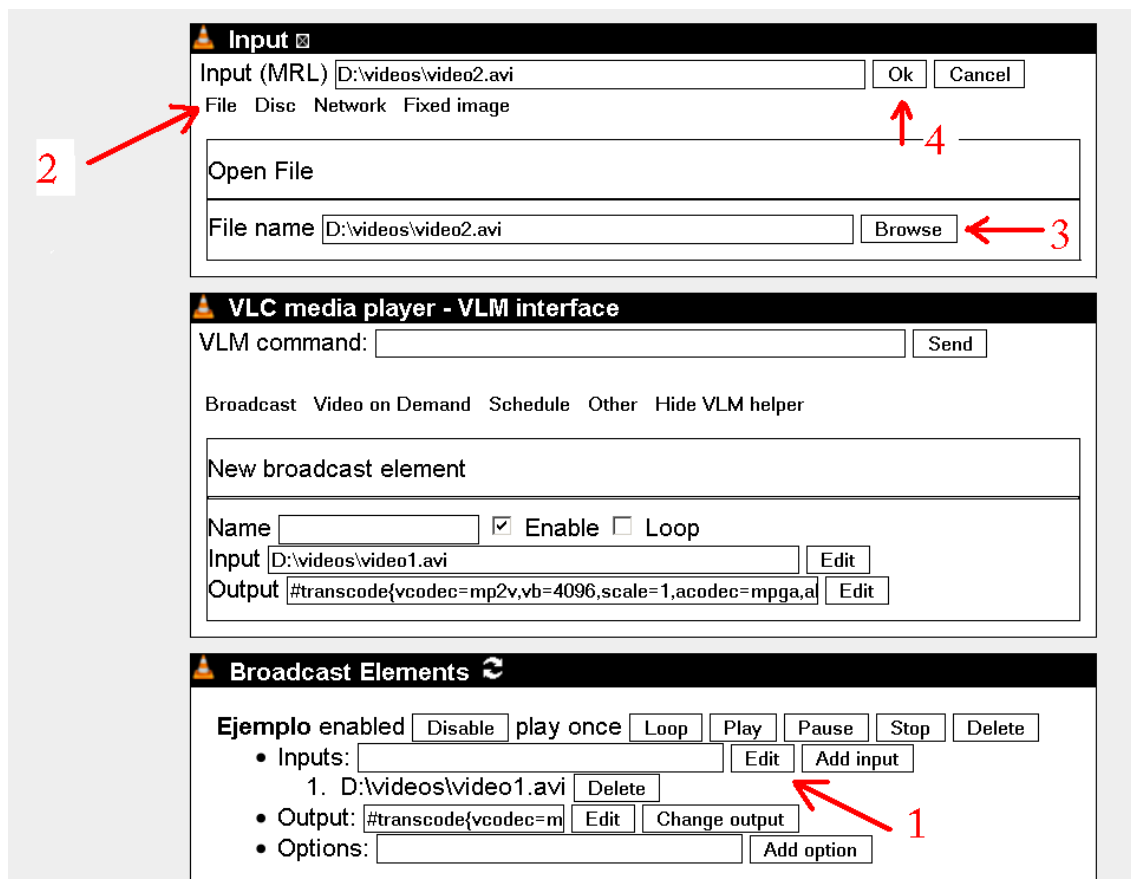


Figura 21

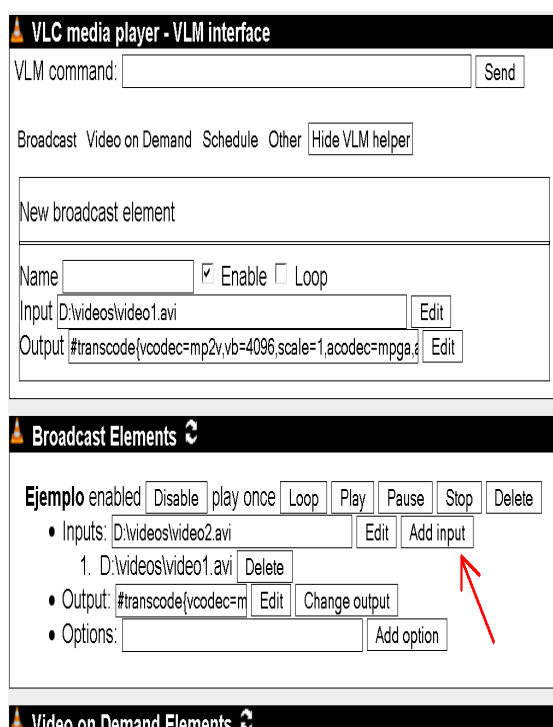


Figura 22

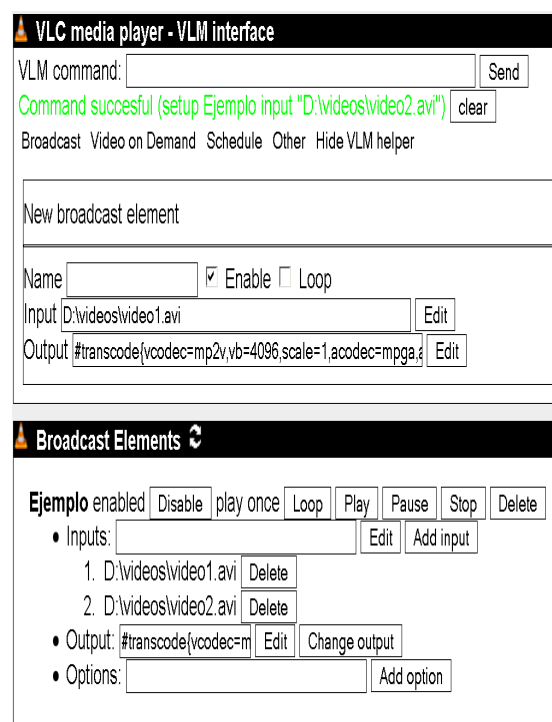
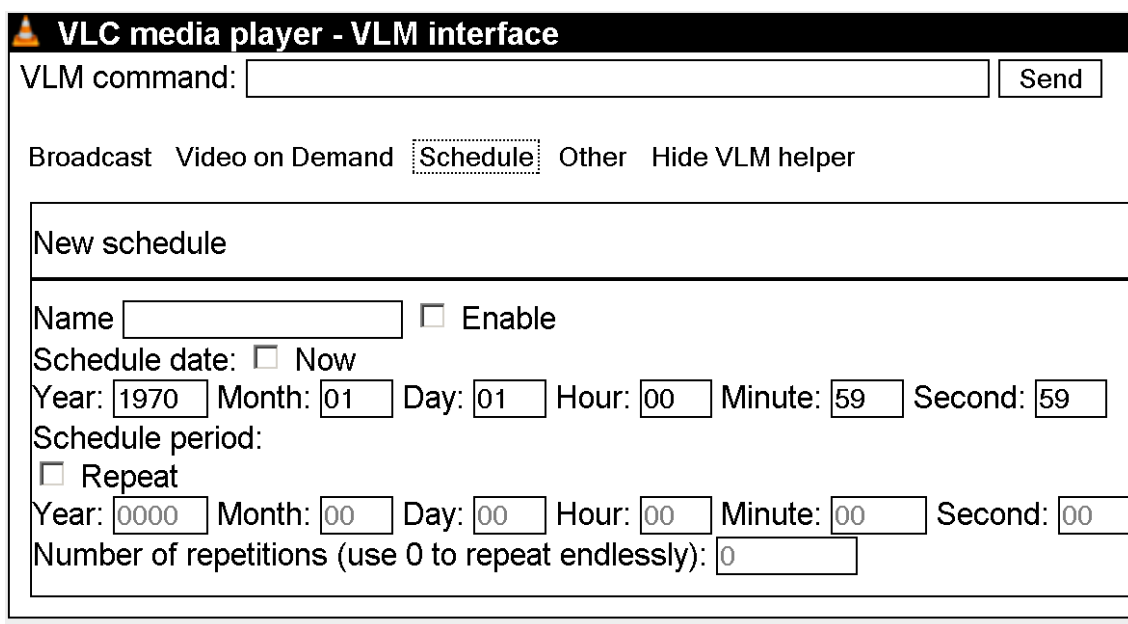


Figura 23

#### 4.2.4.4 Crear elemento de programación

Ahora se pasará a explicar como crear un elemento de programación y como asociarle acciones que actúen sobre un elemento de *broadcast*.

Para crear un elemento de programación se debe pinchar en **Schedule** y una vez ahí (*Figura 24*) se le asigna un nombre y se habilita. Se define la hora de la programación y, si se quiere, el periodo de repetición y el número de repeticiones (si no se especifica un numero de repeticiones se repetirá para siempre con el periodo de repetición establecido). Finalmente se pincha en **send** y queda definido el elemento (*Figura 25*).



**VLC media player - VLM interface**

VLM command:

Broadcast Video on Demand **Schedule** Other Hide VLM helper

---

New schedule

---

Name  ☐ Enable

Schedule date: ☐ Now

Year:  Month:  Day:  Hour:  Minute:  Second:

Schedule period:

☐ Repeat

Year:  Month:  Day:  Hour:  Minute:  Second:

Number of repetitions (use 0 to repeat endlessly):

**Figura 24**



**Broadcast Elements**

Ejemplo enabled  play once

- Inputs:   
  - 1. D:\videos\video1.avi
  - 2. D:\videos\video2.avi
- Output: #transcode{vcodec=m
- Options:

**Video on Demand Elements**

**Schedule Elements**

ejemplo1 enabled

- Date: 2009/12/14-17:59:59
- Period (in seconds): 0/0/1-0:0:0
- Number of repeats left: for ever
- play

**Figura 25**

En la *Figura 25* se ha creado una programación con fecha de 14/Diciembre/2009 a las 17:59:59 que se repetirá cada día a la misma hora sin fin.

Se puede definir las programaciones que se desee, pero para que actúen tenemos que asociarles una acción al menos, bien sea emisión, pausa, parada o búsqueda sobre un elemento de *broadcast* (*Figura 26*). Para ello se elige la acción, se define sobre qué elemento de *broadcast* ha de actuar y se pincha en **Append Command**.

También se pueden asociar dos o más acciones en el mismo elemento de programación. Por ejemplo, una de pausa y otra de emisión sobre dos elementos de *broadcast* diferentes, concatenando de esta manera la emisión de dos elementos de *broadcast* (*Figura 27*).

**Broadcast Elements**

**Ejemplo enabled**








- Inputs:   
  - D:\videos\video1.avi
  - D:\videos\video2.avi
- Output: #transcode{vcodec=m
- Options:

**Video on Demand Elements**

**Schedule Elements**

**ejemplo1 enabled**



- Date: 2009/12/14-17:59:59
- Period (in seconds): 0/0/1-0:0:0
- Number of repeats left: for ever
-

[VLM interface](#) - [VLM interface](#) - [Mosaic wizard](#) - [Flash based remote playback](#) - [VideoLAN website](#)  
 .C media player - version 1.0.3 Goldeneye - (c) 1996-2009 the VideoLAN team

Figura 26

**Broadcast Elements**

**Ejemplo enabled**








- Inputs:   
  - D:\videos\video1.avi
  - D:\videos\video2.avi
- Output: #transcode{vcodec=mp2
- Options:

**Anterior enabled**








- Inputs:   
  - D:\videos\video3.avi
- Output: #transcode{vcodec=mp2
- Options:

**Video on Demand Elements**

**Schedule Elements**

**ejemplo1 enabled**



- Date: 2009/12/14-17:59:59
- Period (in seconds): 0/0/1-0:0:0
- Number of repeats left: for ever
- Command: control Anterior play
-

**ejemplo2 enabled**



- Date: 2009/12/14-18:59:59
- Period (in seconds): 0/0/1-0:0:0
- Number of repeats left: for ever
- Command: control Anterior pause
- Command: control Ejemplo play

Figura 27

En la *Figura 27* se tienen definidos dos elementos de *broadcast*: *Anterior* y *Ejemplo* y también dos elementos de programación: *ejemplo1* y *ejemplo2*.

*Ejemplo* tiene como entradas dos vídeos *vídeo 1* y *vídeo 2* y *anterior* tiene como entrada *vídeo 3*. Lo que se debe hacer es programar el comienzo de *vídeo 3* y una hora mas tarde parar su emisión para que de comienzo *vídeo1* seguido de *vídeo2* y que esto se repita cada día.

Con la programación *ejemplo1* se consigue que la emisión de *Anterior* comience a las 17:59:59 y con la programación *ejemplo 2* se pausa la emisión de *Anterior* a la vez que hacemos que comience la emisión de *Ejemplo* a las 18:59:59.

Si la entrada de *Anterior*, es decir *vídeo3*, durase justamente 1 hora, la emisión se producirá sin ningún corte. En cambio, si dura menos, durante el tiempo desde que *vídeo3* ha terminado y comienza la emisión de *Ejemplo* no se emitirá nada. Si en cambio *vídeo3* dura más de 1 hora al cabo de una hora se pausará su emisión sin finalizar dando paso a *vídeo1* que es la primera entrada de *Ejemplo* una vez termine *vídeo1* seguirá *vídeo2* y una vez termine acabará la emisión.

Co esto se quiere recalcar que a la hora de hacer las programaciones, se debe tener en cuenta la duración de los vídeos si no se quiere que en ciertos momentos no se emita nada. Queda por otro lado la posibilidad de realizar una parrilla continua con un sólo elemento de *broadcast* que contenga todos los vídeos en el orden de emisión deseado y programar su comienzo. Existe una tercera posibilidad y es concatenar en el mismo elemento de *broadcast* un vídeo seguido de un archivo de negro. Así aunque se tenga vídeo se emitirá algo. Con esto se evitará que aparezca "No hay señal" en las televisiones de los usuarios, hasta que comience la emisión del siguiente elemento de *broadcast*.

#### 4.2.4.5 Exportar archivo *VLM*

El archivo *VLM* resultante de esta configuración, se puede observar pinchando en **Other** en la zona principal del *VLM interface* y luego pinchando en **Export VLM Commands** como se puede ver en la *Figura 28*:

**VLC media player - VLM interface**

VLM command:

Broadcast Video on Demand Schedule **Other** Hide VLM helper

#paste your VLM commands here

**Broadcast Elements**

**Ejemplo enabled**  **play once**

- Inputs:
- 1. D:\videos\video1.avi
- 2. D:\videos\video2.avi
- Output: #transcode{vcodec=mp2}
- Options:

**Anterior enabled**  **play once**

- Inputs:
- 1. D:\videos\video3.avi
- Output: #transcode{vcodec=mp2}
- Options:

**Figura 28**

Esto dará lugar a una página donde se encuentra el código de un archivo de configuración *VLM* generado por el ejemplo explicado, con el interfaz web *VLM*.

En la *Figura 29* podemos ver lo que se obtiene:

```
###

##
## VLM HTTP interface
## This file can be loaded as is in VLM.
## Comments starting with "###" were added by the HTTP interface.
## You can remove them if you want to.
##

# VLC media player VLM command batch
# http://www.videolan.org/vlc/

new Ejemplo broadcast enabled
setup Ejemplo input "D:\videos\video1.avi"
setup Ejemplo input "D:\videos\video2.avi"
setup Ejemplo output #transcode{vcodec=mp2v,vb=4096,scale=1,acodec=mpga,ab=128,channels=2}:std{access=udp,mux=ts,dst=224.0.0.1:1234}
new Anterior broadcast enabled
setup Anterior input "D:\videos\video3.avi"
setup Anterior output #transcode{vcodec=mp2v,vb=4096,scale=1,acodec=mpga,ab=128,channels=2}:std{access=udp,mux=ts,dst=224.0.0.1:1234}
new ejemplo1 schedule date 2009/12/17-17:59:59 enabled
setup ejemplo1 period 0/0/1-0:0:0

setup ejemplo1 append control Anterior play
new ejemplo2 schedule date 2009/12/14-18:59:59 enabled
setup ejemplo2 period 0/0/1-0:0:0

setup ejemplo2 append control Anterior pause
setup ejemplo2 append control Ejemplo play

##
## end of export
##
```

**Figura 29**

Además, también se puede importar el *script* de comandos seleccionando **Send Command Batch** una vez copiado el *script* tal y como se indica en la *Figura 28*. Una vez hecho esto, todos los comandos se cargan en la interfaz web. Hay que tener en cuenta que los comandos que se importen de esta manera se añadirán a los ya existentes en la interfaz, con lo cual si lo que se quiere es importar una nueva configuración habrá que borrar primero todos los elementos ya existentes.

## 5 Gestión de los servicios de TDT mediante Opencaster

### 5.1 ¿Que es Opencaster?

*Opencaster* es un conjunto de herramientas de libre distribución enfocadas a generar, procesar, emitir y difundir contenidos encapsulados en *MPEG-2 Transport stream (MPEG-2 TS)* [Mts1]. *MPEG-2 TS* es un protocolo de comunicación para audio vídeo y datos especificados en el estándar *MPEG-2 (ISO/IEC standard 13818-1)*[ISO 13818]. Este protocolo posee unas buenas propiedades en cuanto a la corrección de errores y se utiliza en la emisión de canales digitales de televisión digital terrestre que usan el estándar *DVB-T*.

Con la ayuda de una tarjeta moduladora y *Opencaster* podremos disponer de un canal de televisión digital con servicios avanzados como la guía electrónica de programas (*EPG Electronic Program Guide*) y aplicaciones interactivas.

Las herramientas de *Opencaster* corren en un sistema *Linux* y el *Transport stream* generado puede usarse en distintos soportes de emisión y de datos. Puede ser almacenado en un archivo en memoria, enviado a través de una red multicast dentro de una red IP (para servicios de televisión por IP IPTV) o modulado "al aire" con el uso de una tarjeta moduladora *DVB-T*. Además, el *Transport stream* generado puede ser recibido por decodificadores de bajo coste o televisores con sintonizador *DVB-T* integrado.

Estas herramientas han sido desarrolladas por *Avalpa*. *Avalpa* es una compañía italiana que trabaja en el mercado de la televisión digital [Opc1].

Nuestro caso de estudio será el uso de un sistema *Linux Ubuntu 8.10* [Ubu1], con las herramientas de *Opencaster* instaladas, para la emisión de canales de *DVB-T* con aplicaciones interactivas *MHP*. La emisión será posteriormente modulada con la ayuda de una tarjeta *Dektec DTA 110T* [DTA 110T]. Por ultimo, su recepción se realizará en un decodificador *ADB i-Can 2000T*, simulando así una emisión "al aire".

El objetivo principal de las herramientas *Opencaster* es la adaptación de los *Transport streams* a los requisitos del estándar *DVB-T* para la emisión de televisión digital terrestre. Esta adaptación requiere la multiplexación del vídeo y audio a emitir con la información adicional necesaria para que sea posible su de-multiplexación en los decodificadores. Esta información va incluida en las tablas *PSI-SI (Program Specific Information- Service Information)* [Tps1] del estándar de *MPEG-2*. *Opencaster* permite crear *Transport streams* en las cuales se aloja la información referente a dichas tablas, mediante la ejecución de *scripts* en lenguaje *Python*. Sólo se deberá modificar los *scripts* de la manera adecuada para adaptar las tablas *PSI-SI* a la emisión de nuestros canales de televisión digital terrestre.

*Opencaster* permite también, usando una de sus herramienta, crear los *Transport streams* que alojan las aplicaciones interactivas. Con el uso de

otra herramienta, *Opencaster* será capaz de multiplexar los *Transport streams* de las tablas *PSI* con los *Transport streams* de audio, vídeo y aplicaciones interactivas *MHP* [Mhp1] , para dar lugar a un *Transport stream* completo, apto para la emisión “al aire” y posterior recepción en los decodificadores.

Por último, *Opencaster* posee herramientas que permiten procesar y modificar ciertas características de los *Transport streams*. Además, permite desencapsular los *Transport stream* en *Packetized Elementary Stream (PES)* [Pes1] y estos en *Elementary Stream (ES)* [Els1], disponiendo de herramientas para un análisis más completo de los mismos.



## 5.2 Instalaciones necesarias para el uso de Opencaster

### 5.2.1 Introducción

Para la instalación de *Opencaster* y poder efectuar la emisión es necesario:

- Un ordenador de gama media-alta.
- Un sistema operativo *Linux* basado en *Debian* (en nuestro caso la distribución *Ubuntu*).
- Disponer de conexión a Internet ya que será necesario obtener ciertos paquetes adicionales de la distribución *Ubuntu* para ser capaces de instalar correctamente *Opencaster* y los controladores de la tarjeta moduladora.
- Los archivos de instalación de *Opencaster*.
- Una tarjeta moduladora de *DVB-T* instalada en el ordenador (en nuestro caso una *Dektec DTA 110T*).
- Los controladores de la tarjeta *Dektec*.

A continuación se explicarán los pasos para la instalación de los citados elementos.

### 5.2.2 Instalación del sistema operativo Linux.

En primer lugar se debe obtener la distribución *Linux Ubuntu*. Se ha elegido esta distribución de *Linux* debido a la facilidad de su instalación y actualización. Se puede descargar en su última versión desde la página web [www.ubuntu.com](http://www.ubuntu.com). Una vez descargada es necesario crear un CD de instalación con el archivo *ISO* obtenido. Después de haber grabado el CD de instalación se inserta en la unidad de CD-DVD para proceder a la instalación y se reinicia el ordenador. Siguiendo los pasos mostrados en pantalla se instalará el sistema y una vez finalizado se dispondrá de sistema operativo *Linux* en el ordenador con al menos un usuario con privilegios de administrador.

### 5.2.3 Instalación de Opencaster y los controladores de la Tarjeta Dektec

Una vez instalado y arrancado el sistema *Ubuntu* se encargará automáticamente de la configuración de red si el ordenador está conectado mediante la tarjeta de red *Ethernet* a un router con *DHCP*, en una red de área local.

Así que, en adelante, se supondrá que ya se dispone de una conexión a Internet, la cual es necesaria para los pasos siguientes.

### 5.2.3.1 Descarga de archivos de instalación Opencaster y paquetes Linux necesarios

En primer lugar se deben descargar los archivos de instalación de *Opencaster* de la página web de *Avalpa*, [www.avalpa.com](http://www.avalpa.com). Para ello es necesario registrarse en la página y acceder al área reservada para proceder a la descarga.

También es necesario descargar los controladores de la tarjeta *Dektec* para *Linux*. Esto se consigue desde la dirección:

<http://www.dektec.com/Products/SDK/LinuxSDK/Downloads/LinuxSDK.zip>

Después de descomprimir los archivos descargados en el *Escritorio* ya estarán disponibles todos los archivos de instalación necesarios. Para poder instalarlos y compilarlos en el sistema *Linux*, son necesarios también unos paquetes adicionales que no vienen en la distribución. Para ello *Ubuntu* dispone de una herramienta llamada *gestor de paquetes Synaptic* que se encuentra en las *herramientas de administración* de la pestaña de *Sistema* situada en la barra superior.

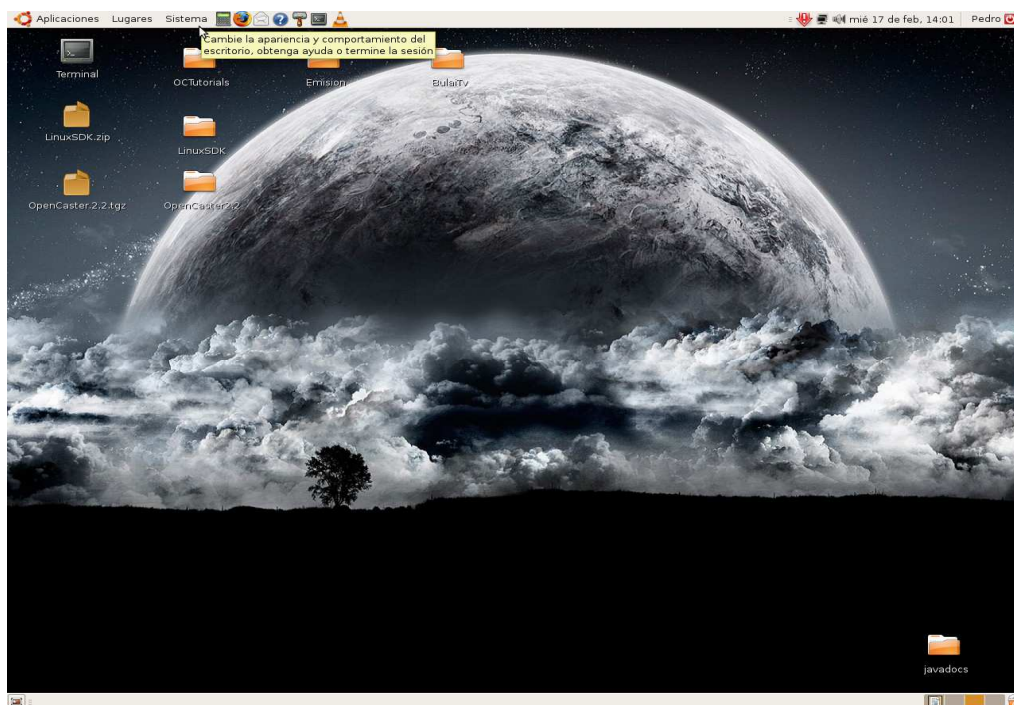


Figura 30

Una vez arrancado el *gestor de paquetes Synaptic*, para lo cual se pide la contraseña de administrador, se procede a marcar para instalar todos los paquetes necesarios para *Opencaster*. Estos paquetes son:

- *binutils*
- *gcc*,
- *gcc-4.3*,
- *libc6-dev*,
- *libgomp1*,

- *Linux-libc-dev,*
- *make,*
- *Python-dev,*
- *Python2.5-dev,*
- *libpcap-dev,*
- *libpcap0.8,*
- *libpcap0.8-dev*
- *zlib1g-dev*

Además, para compilar los controladores de *Dektec* se necesita el paquete `g++`.

Algunos de los paquetes necesarios para *Opencaster* estarán ya disponibles mientras que otros los deberemos marcar para instalar pulsando con el botón derecho del ratón sobre ellos.

Una vez descargados e instalados los paquetes se tendrá todo lo necesario para la instalación y compilación de *Opencaster*, y los controladores de la tarjeta *Dektec*.

### 5.2.3.2 Instalación y compilación de los controladores de la tarjeta *Dektec*

Para proceder a la instalación se deberá abrir una ventana de *Terminal* de *Linux* que es el *shell* para introducir comandos a *Linux*. Para ello se hace click en **Aplicaciones** en la barra superior en la parte izquierda con lo que aparece una lista desplegable donde se debe elegir **Accesorios**. En su lista desplegable aparecerá el *Terminal*.

Con el *Terminal*, para poder instalar *Opencaster* y los controladores, se debe acceder a la carpeta donde se encuentra lo que se vaya a instalar en cada caso. Para acceder a la carpeta donde están situados los controladores de la tarjeta *Dektec* se deberá teclear:

```
equipo@usuario-desktop:~$ cd Escritorio  
equipo@usuario-desktop:~/Escritorio$ cd LinuxSDK  
equipo@usuario-desktop:~/Escritorio/LinuxSDK$ cd Dta1xx  
equipo@usuario-desktop:~/Escritorio/LinuxSDK/Dta1xx$
```

Y para efectuar la instalación de los controladores:

```
equipo@usuario-desktop:~/Escritorio/LinuxSDK/Dta1xx$  
sudo make
```

```
equipo@usuario-desktop:~/Escritorio/LinuxSDK/Dta1xx$  
sudo make install
```

El comando **sudo** sirve para ejecutar comandos con privilegios de administrador y hace que se pregunte por la contraseña del administrador, se usa delante de cualquier comando que requiera dichos privilegios. Por ejemplo, para instalar paquetes nuevos a través de línea de comandos.

Ahora estarán instalados los controladores pero para poder usar la tarjeta se deben inicializar. La operación de inicialización de los controladores de la tarjeta *Dektec* debe ser ejecutada cada vez que se arranque el sistema *Linux*. Para poder usar el comando de inicialización en cualquier momento, sin estar en la carpeta de los *drivers*, será preciso ejecutar el siguiente comando:

```
equipo@usuario-desktop:~/Escritorio/LinuxSDK/Dta1xx$  
sudo install -m 755 Dta1xxInit /usr/local/bin
```

El comando anterior funciona de la siguiente manera: Se usa *sudo* para tener privilegios de administrador para instalar el comando de inicialización; *-m 755* es el modo en el que será instalado (permisos de lectura escritura y ejecución para los usuarios); se especifica el comando a instalar y por ultimo se especifica el directorio donde será instalado. El directorio, en este ejemplo, es el directorio donde los usuarios suelen instalar los programas en el sistema operativo *Linux*, para poder ejecutarlos en cualquier *path*.

Para proceder a la inicialización de la tarjeta se debe teclear en el *Terminal* de *Linux* el siguiente comando con privilegios de administrador:

```
sudo Dta1xxInit start
```

Para poder usar la tarjeta *Dektec* también se precisa instalar una herramienta cuya misión es lanzar la emisión con los parámetros de modulación deseados. Esta herramienta es el ***DtPlay***.

Nuevamente se debe compilar la herramienta y, para usarla con comodidad, instalarla de forma que pueda ser ejecutada en cualquier *path*. Para proceder a ello se deberá estar en la carpeta del ***DtPlay***:

```
equipo@usuario-desktop:~/Escritorio/LinuxSDK$ cd DtPlay  
equipo@usuario-desktop:~/Escritorio/LinuxSDK/DtPlay$ sudo make  
equipo@usuario-desktop:~/Escritorio/LinuxSDK/DtPlay$  
sudo install -m 755 DtPlay /usr/local/bin
```

De nuevo se instala con privilegios de administrador y permisos de lectura escritura y ejecución el comando *DtPlay* en el directorio */usr/local/bin*. De esa forma podrá ser ejecutado en cualquier directorio.

Si el comando de inicialización ha sido ejecutado ya se dispondrá de la tarjeta *Dektec* totalmente operativa.

### 5.2.3.3 Instalación y compilación de *Opencaster*

Para la instalación de los comandos de *Opencaster* también es necesario estar situados en la carpeta donde se hayan descomprimido los archivos de instalación.

```
equipo@usuario-desktop:~$ cd Escritorio  
equipo@usuario-desktop:~/Escritorio$ cd Opencaster2.2
```

```
equipo@usuario-desktop:~/Escritorio/Opencaster2.2$  
sudo make  
equipo@usuario-desktop:~/Escritorio/Opencaster2.2$  
sudo make install
```

Así, estarán instaladas y operativas todas las herramientas de *Opencaster* en cualquier *path*.

## 5.3 Uso de Opencaster

### 5.3.1 Introducción

- **Multiplexación de un *Transport stream* para emisión en DVB-T**

Para que sea posible la emisión de contenidos audiovisuales en el estándar *DVB-T* se debe multiplexar varios tipos de información:

- El vídeo de la emisión de un canal de televisión digital (servicio).
- El audio de la emisión del servicio.
- La información de las tablas *PSI-SI* de multiplexación del estándar *DVB-T* que definen la estructura del *Transport stream* multiplexado para la emisión.
- Opcionalmente se puede incluir un flujo de datos (Carrusel de objetos) que alojarán las aplicaciones interactivas.
- Flujo de paquetes sin información para rellenar el *Transport stream* y tener la tasa de bits requerida por los parámetros de modulación *DVB-T* escogidos.

Cada tipo de información en el *Transport stream* multiplexado será otro *Transport stream* que deberá llevar un identificador asociado, llamado *PID*.

A la hora de emitir un contenido audiovisual en televisión digital es muy importante definir de manera correcta las tablas del estándar *DVB-T*.

- **Tablas *PSI-SI* de una emisión en *Transport stream* en DVB-T**

A continuación se describirán las tablas más importantes del estándar *DVB-T* [Tablas *PSI-SI*].

- ***PAT (Program Asociation Table)***: Tiene *PID 0*. Sirve como un índice que indica qué *PID*'s tienen las tablas *PMT* de cada servicio de televisión digital en un *Transport stream* multiplexado.
- ***PMT (Program Map Table)*** de un servicio de televisión digital: Tiene la *PID* descrita en la *PAT*. Indica qué *PID*'s están asociadas a un servicio así como el tipo de contenido de los mismos.
- ***NIT (Network Information Table)***: Tiene *PID 16*. Describe la red física usada para transmitir el *Transport stream*.
- ***SDT (Service Description Table)***: Tiene *PID 17*. Describe los servicios de televisión digital del *Transport stream* y sus parámetros tales como el nombre del servicio, el proveedor del servicio, etc.
- ***EIT (Event Information Table)***: Tiene *PID 18*. Describe los eventos que se van a producir, hora de comienzo, duración, información que se muestra al usuario; se suele usar para la información del *EPG*.
- ***AIT (Aplication Information Table)***: Tiene el *PID* descrito en la *PMT* del servicio. Describe las aplicaciones interactivas existentes en el *Transport stream* y toda la información necesaria para su ejecución correcta en el decodificador.

- **TDT(Time Description Table)**: Tiene *PID* 20. Describe la fecha y hora actual en *UTC*.

Las tablas cuyos *PID*'s no están definidos en el estándar son la *PMT* y la *AIT*, así como los *PID*'s del vídeo, del audio y del carrusel de objetos que lleva los datos para las aplicaciones. Todos ellos varían al estar asociadas a los distintos servicios de televisión digital.

El *PID* del *Transport stream* de paquetes vacíos viene definido también por el estándar *DVB-T* y es el *PID* 8191.

*Opencaster* posee 2 partes diferenciadas a la hora de trabajar con él:

- *Scripts* en lenguaje *Python* para la creación de tablas *PSI-SI*
- Comandos *Opencaster*

- ***Scripts* en lenguaje *Python***

Por un lado *Opencaster* hace uso de *scripts* en lenguaje *Python*. El lenguaje *Python* es un lenguaje interpretado, que facilita la creación de *scripts* por sus características, además, es orientado a objetos.

Los *scripts* de *Opencaster* serán usados para la creación de los *Transport streams* que alojaran las tablas *PSI-SI* del estándar *MPEG-2*. Dichos *Transport streams* de las tablas son necesarios para que los decodificadores sean capaces de interpretar el *Transport stream* multiplexado emitido y posteriormente de-multiplexar los distintos servicios de televisión digital y mostrarlos por pantalla.

*Opencaster* permite modificar los parámetros de las tablas y crear un *Transport stream* para cada una de ellas. Estos *Transport streams* que alojan las tablas deberán ser multiplexados con los *Transport streams* de vídeo, de audio y del carrusel de objetos donde se hallan las aplicaciones, para obtener un *Transport stream* apto para la emisión.

Para crear los *Transport streams* que alojan las tablas, *Opencaster* hace uso de *scripts* escritos en lenguaje de programación *Python*, como se ha comentado. Modificando el *script* podremos definir parámetros de las tablas como las *PID*'s de las *PMT*'s de los servicios a emitir así como las *PID*'s de su audio y vídeo, describir las aplicaciones interactivas que incluiremos en el *Transport stream*, añadir eventos para la *EPG* (Guía electrónica de Programas), etc.

Una vez modificado el *script*, deberá ser ejecutado para crear los *Transport streams* de las tablas que posteriormente multiplexaremos con el audio, el vídeo, las aplicaciones y el *Transport stream* de paquetes vacíos.

- **Comandos *Opencaster***

Por otro lado *Opencaster* posee herramientas en forma de comandos que deberán ser ejecutados en la *shell* de *Linux*. Estos comandos sirven para



diversos propósitos. El más importante se utiliza para la multiplexación de varios *Transport streams* en uno solo. Este es el más importante ya que servirá para multiplexar los contenidos a emitir, con las tablas *PSI-SI* necesarias para su correcta recepción por parte de los decodificadores de televisión digital terrestre.

También las herramientas de *Opencaster* se usan para el análisis de *Transport streams*, además de permitir transformarlos en *PES* y estos a su vez en *ES* para su análisis. Otro uso de las herramientas de *Opencaster* es el procesado de los *Transport streams*, permitiendo modificar diversas propiedades de los mismos.

- **Puesta en emisión de un servicio de televisión digital con *Opencaster* y VLC.**

Para el objetivo marcado de poner en emisión un servicio de televisión digital con *Opencaster* se siguen los siguientes pasos:

En primer lugar se modifica y ejecuta un *script* en *Python* de la manera adecuada para crear los *Transport stream* de las tablas *PSI-SI* necesarias para la emisión de un cierto servicio. También es posible poner varios servicios en emisión en el mismo *Transport stream* multiplexado.

En segundo lugar, con la ayuda de una herramienta de *Opencaster*, se crea el *Transport stream* que aloja las aplicaciones interactivas.

En tercer y último lugar se usan otras herramientas de *Opencaster* para:

- La recepción de los contenidos a emitir (audio y vídeo del servicio), enviados por el programa *VLC*.
- Procesado de los contenidos para su multiplexación con las tablas *PSI-SI*
- Multiplexación de los contenidos con las tablas *PSI-SI* y el *Transport stream* que aloja las aplicaciones.
- Corrección del *Transport stream* multiplexado para su correcta recepción en los decodificadores.
- Corrección de la información de la fecha y hora del *Transport stream* ordenador

Finalmente se utiliza una herramienta propia de la tarjeta moduladora para proceder a la modulación del *Transport stream* multiplexado y corregido para su emisión.

En el siguiente apartado se describirá un *script* en *Python* para la generación de las tablas *PSI-SI*, comentando su funcionamiento y partes más importantes

### 5.3.2 Descripción del Script usado por Opencaster para la creación de los Transport stream de las tablas PSI-SI

A continuación se verá un ejemplo de un *script* en *Python* para la creación de los *Transport streams* que alojan tablas *PSI-SI* y se comentarán sus partes más importantes y su funcionamiento [Opc2].

El *script* tiene 4 partes diferenciadas:

La primera parte incluye todas las librerías que se usaran para la correcta ejecución del *script*. Estas librerías forman parte de *Opencaster* y contienen las definiciones de las clases que se usarán para crear las secciones referentes a cada una de las tablas *PSI-SI* a la hora de ejecutar el *script*.

Las secciones son estructuras sintácticas en lenguaje binario donde se recoge la información y los parámetros de las tablas *PSI-SI* de manera que se pueda encapsular en los *Transport streams*. Con ayuda del *script* se completarán los parámetros que se necesiten definir. Cada sección correspondiente a cada una de las tablas, tienen una sintaxis concreta definida por el estándar de *MPEG-2 TS* recogido en documento ISO/IEC 13818. El *script* utiliza las librerías de *Opencaster* para crear adecuadamente las secciones siguiendo su sintaxis con los parámetros que se introducirán.

En la segunda parte se crearán ciertas variables que se usarán en las siguientes partes del *script*, y que facilitan su edición.

La tercera parte consta de la definición de los parámetros de las distintas secciones, correspondientes a cada una de las tablas necesarias en el *Transport stream* que se multiplexará.

Por último, la cuarta parte se ocupa de las instrucciones necesarias para la creación de las secciones de las tablas, con los parámetros definidos en la tercera parte y su transformación en los *Transport streams* individuales correspondientes a las tablas.

En el ejemplo comentado se crearán los *Transport stream* correspondientes a las tablas *PSI-SI* necesarios para multiplexar un servicio de televisión digital con una aplicación *MHP* embebida en un carrusel de objetos *DSM-CC* [Dsm1] y con 2 eventos programados a ciertas horas.

#### 5.3.2.1 Primera Parte: Librerías.

Se observa, a continuación las librerías, que importa el *script*. En ellas están definidas las clases usadas para crear cada una de las secciones referentes a las tablas *PSI-SI*. Cada librería hace uso a su vez de otras librerías para la definición de las clases. Se pueden observar cada una de las librerías en el directorio *libs* del directorio donde se descomprimió *Opencaster*.

```
#!/usr/bin/env Python  
import os
```

```
from dvbobjects.PSI.PAT import *
from dvbobjects.PSI.NIT import *
from dvbobjects.PSI.SDT import *
from dvbobjects.PSI.PMT import *
from dvbobjects.PSI.TDT import *
from dvbobjects.PSI.EIT import *
from dvbobjects.DVB.Descriptors import *
from dvbobjects.MPEG.Descriptors import *
from dvbobjects.MHP.AIT import *
from dvbobjects.MHP.Descriptors import *
```

### 5.3.2.2 Segunda Parte: Variables.

#Para facilitar el uso del *script* se crean unas variables que se usaran en las dos siguientes partes del mismo.

**Avalpa\_transport\_stream\_id = 1**

#El *Transport stream* multiplexado debe tener un identificador único; 1 es un valor para demostración ya que el *ID* del *Transport stream* debería ser solicitado a la organización *DVB*.

**Avalpa\_original\_transport\_stream\_id = 1**

#1 es un valor para demostración un valor para este parámetro debería ser solicitado a la organización *DVB*.

**Avalpa1\_service\_id = 1**

#Cada servicio de televisión digital debe tener un identificador único; 1 es un valor para demostración ya que el *ID* del servicio debe ser solicitado a la organización *DVB*.

**Avalpa1\_PMT\_pid = 1031**

#Este será el valor que se usará para el *PID* de la *PMT* del servicio de televisión digital, si se quisiera tener más servicios en el *Transport stream* se deberán definir otros *PID*'s para sus *PMT*'s.

**ait1\_pid = 2001**

# Este será el valor que se va a utilizar para el *PID* de la *AIT* del servicio de televisión digital.

**dsmccB\_association\_tag = 0xB**

#Este es un valor relacionado con el *Transport stream* de la aplicación interactiva, mas adelante cuando se cree el *Transport stream* de la aplicación con la ayuda de las herramientas de *Opencaster*, se usará este valor para asociarlo.

**dsmccB\_carousel\_id = 1**

#Este también es un valor relacionado con el *Transport stream* de la aplicación interactiva, mas adelante cuando se cree dicho *Transport stream*, se usará este valor para asociarlo.

### 5.3.2.3 Tercera Parte: Secciones de las tablas.

Se debe comentar que para cada sección se va a crear un objeto de la clase correspondiente a la sección particular que va a crear en ese momento. Por ejemplo a continuación se observa que se va a crear un objeto llamado *nit* de la clase *network\_information\_section*. Más adelante se verá como para las diferentes secciones se crean objetos similares pero de distintas clases.

Cada clase correspondiente a cada sección estará definida de distinta forma ya que corresponde a diferentes secciones y cada sección tiene una sintaxis y unos parámetros diferentes. Además, para cada sección estarán definidas otras clases que se usaran de manera particular para crear dicha sección y que incluirán los parámetros que se deben definir para las secciones de las tablas.

# En primer lugar se introducirán los parámetros para la sección de la tabla **NIT Network Information Table**[Nit1].

#Se crea el objeto *nit* que es de la clase *network\_information\_section* que es una subclase de la clase *section*

**nit = network\_information\_section(**

# Se le da un identificador a la red.

**network\_id = 1,**

**network\_descriptor\_loop = [**

#Aquí queda definido el nombre de la red.

**network\_descriptor(network\_name = "UPNA"),**

**],**

# Se procede a definir el *Transport stream* que aloja la emisión de los servicios.

**transport\_stream\_loop = [**

**transport\_stream\_loop\_item(**

#Aquí se define el identificador del *Transport stream* de la emisión.

**transport\_stream\_id = Avalpa\_transport\_stream\_id,**

**original\_network\_id=Avalpa\_original\_transport\_stream\_id,**

**transport\_descriptor\_loop = [**

**service\_list\_descriptor(**

**dvb\_service\_descriptor\_loop = [**

#Se informa que el *Transport stream* lleva un servicio de televisión digital con cierto *service\_ID*. Se podrían añadir más elementos *service\_descriptor\_loop\_item* completos con otros *service\_ID* para informar que se van a añadir más servicios de televisión digital en el *Transport stream*.

**service\_descriptor\_loop\_item(**

#Se define el identificador del servicio de televisión digital.

**service\_ID = Avalpa1\_service\_id,**

#El tipo es 1 que significa que es un servicio de televisión digital.

**service\_type = 1,**

**),**

**],**

**),**

**],**

**),**

**],**

#Se debe cambiar el numero de versión cada vez que se modifique esta tabla para que el decodificador se de cuenta de que ha cambiado y actualice la tabla.

**version\_number = 1,**

**section\_number = 0,**

**last\_section\_number = 0,**

**)**

#Ahora se verá la sección de la **PAT Program Association Table** [Pat1].

#De nuevo se crea el objeto *pat* que es de la clase *program\_asociation\_section* que es una subclase de la clase *section*.

**pat = program\_association\_section(**

#Se define el *ID* del *Transport stream* para el cual la *PAT* va a ser creada.

**transport\_stream\_id = Avalpa\_transport\_stream\_id,**

**program\_loop = [**

#Se definirá un servicio de televisión digital al que se debe asignar su *ID* y definirle el *PID*, si se tuviese otro servicio más se deberá crear su correspondiente *program\_loop\_item* con su *ID* y su *PID* asociado para el *Transport stream*.

**program\_loop\_item(**

#El *servicio de television digital* debe tener un identificador proporcionado por la organización *DVB*.

**program\_number = Avalpa1\_service\_id,**

#En la siguiente línea queda definido el *PID* de la *PMT* del servicio, en este caso *1031*, ya que al definir las variables auxiliares se le asignó este número.

**PID = Avalpa1\_PMT\_pid,**

**),**

#Si el *Transport stream* lleva la tabla *NIT* se debe crear un *program\_loop\_item* especial dicha tabla.

**program\_loop\_item(**

#En esta parte el *program\_number = 0* indica que se refiere a la *NIT*. La *NIT* debe tener *PID 16* según el estándar.

**program\_number = 0,**

**PID = 16,**

**),**

**],**

#Se debe cambiar el numero de versión cada vez que se modifique esta tabla para que el decodificador se de cuenta de que ha cambiado y actualice

la tabla.

```
version_number = 1,  
  
section_number = 0,  
  
last_section_number = 0,  
  
)
```

#Ahora se comentará la sección del **SDT Service Description Table** [Sdt1]. Esta es una *SDT* básica con los *descriptores* mínimos necesarios, *Opencaster* posee una librería para añadir más posibles datos en la *SDT*.

#Como en las secciones anteriores se crea un objeto *sdt* del tipo *service\_description\_section*.

```
sdt = service_description_section(
```

#Se define el *ID* del *Transport stream* para el que se está definiendo la tabla *SDT*.

```
transport_stream_id = Avalpa_transport_stream_id,  
  
original_network_id = Avalpa_original_transport_stream_id,
```

#A continuación se describen los servicios del *Transport stream* en el *service loop*.

```
service_loop = [
```

#Cada servicio de televisión digital debe tener un *service\_loop\_item* que definirá el servicio. Se debe identificar para qué servicio se están definiendo sus parámetros con el *service\_ID*.

```
service_loop_item(  
  
service_ID = Avalpa1_service_id,
```

#Con el parámetro siguiente se indicará si el servicio lleva información de eventos de programación en la tabla *EIT*. Si ponemos un 0 no se están emitiendo y si ponemos un 1 si se están emitiendo.

```
EIT_schedule_flag = 1,
```

#Con la opción siguiente se indica si actualmente se tiene información de eventos del tipo *present\_following* en la tabla *EIT*. Si se pone un 0 no se están emitiendo y si se pone un 1 si se están emitiendo.

```
EIT_present_following_flag = 1,
```



#Aquí se informa si el servicio se esta emitiendo actualmente (4), no se está emitiendo (1), empieza en unos segundos (2) o esta pausado (3).

**running\_status = 4,**

#Este parámetro informa si el servicio esta codificado (1) ó no (0). A continuación se describe el nombre del proveedor y del sevicio de televisión digital.

**free\_CA\_mode = 0,**

#Se procede a describir el servicio

**service\_descriptor\_loop = [**

**service\_descriptor(**

#Se define el tipo de servicio, proveedor y nombre del servicio.

**service\_type = 1, # digital television service**

**service\_provider\_name = "UPNA",**

**service\_name = "CANALTV",**

**),**

**],**

**),**

**],**

**version\_number = 1,**

#Se debe cambiar el numero de versión cada vez que se modifique esta tabla para que el decodificador se de cuenta de que ha cambiado y actualice la tabla.

**section\_number = 0,**

**last\_section\_number = 0,**

**)**

#En siguiente lugar se verá la sección del **PMT Program Map Table**[Pmt1]. En este caso el **PMT** posee **descriptores** de la **AIT** para aplicaciones interactivas con el estándar **MHP**. Si tuviéramos otro servicio se deberá definir otra **PMT** para el otro servicio con todos sus parámetros.

#Se crea un objeto de la clase *program\_map\_section* que es una subclase

de la clase *section*.

**pmt = program\_map\_section(**

#Se identifica el servicio para el cual se esta definiendo la *PMT* con el *service\_ID*.

**program\_number = Avalpa1\_service\_id,**

#Se indica en qué *PID* está la información de los *PCR* (las referencias del reloj, habitualmente insertadas en el vídeo).

**PCR\_PID = 2064,**

**program\_info\_descriptor\_loop = [],**

#Se deben definir los contenidos de cada servicio para ello se crea un *stream\_loop\_item* para cada contenido.

**stream\_loop = [**

**stream\_loop\_item(**

#El tipo de flujo es 2 que indica que es un flujo de vídeo en *MPEG-2* [Pmt1].

**stream\_type = 2,**

#Se indica que el vídeo tendrá como *PID* 2064.

**elementary\_PID = 2064,**

**element\_info\_descriptor\_loop = []**

**),**

**stream\_loop\_item(**

#El tipo de flujo es 3 que indica que es un flujo de audio en *MPEG-2*[Pmt1].

**stream\_type = 3,**

#Se indica que el audio tendrá como *PID* 2068.

**elementary\_PID = 2068,**

**element\_info\_descriptor\_loop = []**

**),**

**stream\_loop\_item(**

#El tipo de flujo es 5 que indica que es un flujo de tipo sección privada[Ait2]. Este flujo se utilizara para alojar la tabla *AIT* .

**stream\_type = 5,**

#Se indica que la tabla *AIT* tendrá como *PID* *ait1\_pid = 2001*.

**elementary\_PID = ait1\_pid,**

**element\_info\_descriptor\_loop = [**

**application\_signalling\_descriptor(**

**application\_type = 1,**

# Se indica que la aplicación es del tipo *DVB-J* con 1, si tuviéramos una aplicación tipo *DVB-HTML* deberíamos indicar 2 [Ait1].

**AIT\_version = 1,**

#La actual versión de *AIT*.

**),**

**]**

**),**

**stream\_loop\_item(**

#El tipo de flujo es 11 que indica que es un flujo de tipo carrusel de objetos *DSMCC*, (es un estándar para transmisión de datos a través en emisiones de televisión digital de *MPEG-2*)[Pmt1].

**stream\_type = 11,**

#Se indica que el flujo de datos del carrusel tendrá como *PID* 2003.

**elementary\_PID = 2003,**

**element\_info\_descriptor\_loop = [**

#A continuación se usan varios *descriptores* que especifican las propiedades del flujo de datos del carrusel de objetos *DSMCC*.

**association\_tag\_descriptor(**

#Esta marca de asociación identifica el flujo de datos en carrusel de objetos, se usara cuando creamos el *Transport stream* que contiene las

aplicaciones y en la tabla *AIT*.

```
association_tag = dsmccB_association_tag,  
use = 0,
```

#Los siguientes valores son por defecto y no se pueden cambiar.

```
selector_lenght = 0,  
transaction_id = 0x80000000,  
timeout = 0xFFFFFFFF,  
private_data = "",  
),
```

```
stream_identifier_descriptor(
```

#Esto es lo mismo que la marca de asociación. Algunos decodificadores usaran la *association\_tag* mientras que otros usaran la *component\_tag*.

```
component_tag = dsmccB_association_tag,  
),
```

```
carousel_identifier_descriptor(
```

#Este es el identificador del carrusel, tiene un propósito similar a los anteriores que es identificar el flujo de datos del carrusel de objetos.

```
carousel_ID = dsmccB_carousel_id,  
format_ID = 0,  
private_data = "",  
),
```

```
data_broadcast_id_descriptor(
```

#240 es el código que especifica que el carrusel *DSMCC* es un carrusel del estándar *MHP*[Ait3].

```
data_broadcast_ID = 240,  
ID_selector_bytes = "",  
),
```

```
]
)
],
version_number = 1,
```

#Se debe cambiar el numero de versión cada vez que se modifique esta tabla para que el decodificador se de cuenta de que ha cambiado y actualice la tabla.

```
section_number = 0,
last_section_number = 0,
)
```

#Si se dispusiera de más servicios se debe crear otro objeto de la clase *program\_map\_section* para cada uno de ellos que describan los flujos de cada servicio y sus *PID's*.

#La siguiente sección que se va a describir se trata de la correspondiente a la tabla *AIT*, ***Application Information Table***[Ait1]. En ella se definen las propiedades de las aplicaciones que irán en el *Transport stream*.

#De nuevo se crea un objeto llamado *ait* de la clase *application\_information\_section* que es una subclase de la clase *section*.

```
ait = application_information_section(
```

#Comienza la descripción de una aplicación, podríamos incluir otra aplicación en la misma *AIT* o bien en otra *AIT* para otro servicio.

```
application_type = DVB_J_application_type,
common_descriptor_loop = [],
application_loop = [
    application_loop_item(
```

#La aplicación debe tener el identificador de la empresa que la ha creado. Este valor es un valor de demostración, deberíamos pedir un *ID* a la organización *DVB*.

```
organisation_id = 10,
```

#Es el identificador de la aplicación, debería ser única porque identifica a la aplicación.

**application\_id = 1001,**

#Se dispone de varias opciones a la hora de lanzar la aplicación, si se indica 1 la aplicación comenzará automáticamente, si se indica 2 la aplicación se añadirá a la lista de aplicaciones disponibles para ejecutar, si se indica 3 se señalizará a la aplicación para que pare y si se indica 4 parará la ejecución de la aplicación.

**application\_control\_code = 2,**

**application\_descriptors\_loop = [**

**transport\_protocol\_descriptor(**

#La aplicación se emite en un carrusel de objetos *MHP DSMCC*.

**protocol\_id = MHP\_OC\_protocol\_id,**

#Aquí se debe poner el *ID* del carrusel.

**transport\_protocol\_label = 1,**

**remote\_connection = 0,**

#Aquí se debe poner el *asociation\_tag*.

**component\_tag = 0xB,**

**),**

**application\_descriptor(**

# Se debe informar de la versión *MHP* que usa la aplicación en este caso *MHP 1.0.2*.

**application\_profile = 0x0001,**

**version\_major = 1,**

**version\_minor = 0,**

**version\_micro = 2,**

**service\_bound\_flag = 1,**

# La aplicación es visible al usuario.

**visibility = 3**

```
        application_priority = 1,

#ID del carrusel.

        transport_protocol_labels = [1],

    ),

# El nombre de la aplicación.

    application_name_descriptor(application_name = "Elementos
Havi"),

#Los parámetros que se le pasan al xlet.Si se le indican parámetros éstos
deben ir entre comillas dentro de los corchetes

    dvb_j_application_descriptor(parameters = []),

    dvb_j_application_location_descriptor(

#Directorio raíz dentro de la estructura del carrusel de objetos.

        base_directory = "/",

#Podemos añadir un path en el carrusel.

        class_PATH_extension = "",

# La clase principal de la aplicación que ejecuta el Xlet.

        initial_class = "ElementosHavi",

    ),

]

),

],

version_number = 1,

section_number = 0,

last_section_number = 0,

)

#La siguiente sección que tenemos es la de la tabla TDT (Time
Description Table)[Tdt1].
```



#Se crea un objeto tdt de la clase *time\_date\_section* que es una subclase de la clase *section*.

**tdt = time\_date\_section(**

#Se pone el año con referencia a 1900, 1900+109=2009.

**year = 109,**

**month = 11,**

**day = 16,**

#Se pone, por ejemplo, las 11:30:21 y se usan los hexadecimales como decimales.

**hour = 0x11,**

**minute = 0x30,**

**second = 0x21,**

**version\_number = 1,**

**section\_number = 0,**

**last\_section\_number = 0,**

**)**

#La última sección que se va a describir es la correspondiente a la tabla *EIT* **Event Information Table**[Eit1] En el ejemplo se lanzarán 2 eventos uno actual y otro que sigue al actual.

#Se crea un objeto llamado *eit* de la clase *event\_information\_section* que es una subclase de la clase *section*.

**eit = event\_information\_section(**

**table\_id = EIT\_ACTUAL\_TS\_PRESENT\_FOLLOWING,**

#Se identifica el *Transport stream* que aloja la tabla *EIT* y a qué servicio corresponde la información del evento.

**service\_id = Avalpa1\_service\_id,**

**transport\_stream\_id = Avalpa\_transport\_stream\_id,**

**original\_network\_id = Avalpa\_original\_transport\_stream\_id,**

**event\_loop = [**

#Aquí comienza la descripción del evento, la hora de comienzo, duración, e información que contiene.

**event\_loop\_item(**

#El evento tiene como identificador el 1.

**event\_id = 1,**

**start\_year = 110,**

**start\_month = 2,**

**start\_day = 9,**

#Hay que tener cuidado a la hora de poner la hora de comienzo ya que debemos poner la hora -1 por ejemplo para que empiece a las 10 hay que poner las 9.

**start\_hours = 0x09,**

**start\_minutes = 0x05,**

**start\_seconds = 0x00,**

**duration\_hours = 0x04,**

**duration\_minutes = 0x00,**

**duration\_seconds = 0x00,**

#4=El servicio esta emitiendo, 1= no emitiendo, 2= comienza en unos segundos, 3= pausado.

**running\_status = 4,**

#Acceso condicional 0= no codificado 1=si codificado.

**free\_CA\_mode = 0,**

**event\_descriptor\_loop = [**

#Se describen las propiedades del evento: nombre y texto que incluye

**short\_event\_descriptor (**

**ISO639\_language\_code = "ita",**

**event\_name = "Demo",**

#A la hora de incluir un texto no se puede alargar demasiado ya que se tiene una longitud máxima para la sección de esta tabla de 4096 bytes.

```
        text = "Texto de información",  
    )  
    ],  
),  
],  
version_number = 1,  
section_number = 0,
```

#Como se tiene otra sección de *EIT* la última sección será la 1.

```
last_section_number = 1,  
)
```

#Se crea otro objeto de la clase *event\_information\_section* de nombre *eit\_follow* de la misma forma que el anterior pero a otra hora y con otro texto.

```
eit_follow = event_information_section(  
    table_id = EIT_ACTUAL_TS_PRESENT_FOLLOWING,  
    service_id = Avalpa1_service_id,  
    transport_stream_id = Avalpa_transport_stream_id,  
    original_network_id = Avalpa_original_transport_stream_id,  
    event_loop = [  
        event_loop_item(  
            event_id = 2,  
            start_year = 110,  
            start_month = 2,  
            start_day = 9,
```

```
start_hours = 0x15,  
start_minutes = 0x05,  
start_seconds = 0x00,  
duration_hours = 0x12,  
duration_minutes = 0x00,  
duration_seconds = 0x00,  
running_status = 1,  
free_CA_mode = 0,  
event_descriptor_loop = [  
    short_event_descriptor (  
        ISO639_language_code = "ita",  
        event_name = "Demo terminada",  
        text = "Texto de Información del siguiente contenido",  
    )  
],  
),  
],  
version_number = 1,
```

#Ahora se tiene la segunda sección que tiene como número el 1.

```
section_number = 1,  
last_section_number = 1,  
)
```

#### 5.3.2.4 Cuarta parte: Creación de los *Transport streams*

Por último, en el *script* se ejecutan varias órdenes que se ocupan de, a partir de los objetos definidos anteriormente, crear los archivos correspondientes a las secciones de las tablas y a partir de dichos archivos, los *Transport streams* correspondientes a cada una de las tablas.

Se debe comentar que para todas y cada una de las clases correspondientes a las distintas secciones de las tablas existe un método llamado *pack*. La función del método *pack* es transformar los objetos en las secciones con la sintaxis correcta de la sección e incluir todos los parámetros definidos en la creación de los objetos. Una vez transformados los objetos en las secciones el *script* guardará la sección en un archivo con extensión ".sec". A partir de los archivos ".sec" el *script* hará uso de una herramienta de *Opencaster* que se ocupa de transformar las secciones en paquetes *Transport stream*. Esta herramienta es **sec2ts** y funciona de la siguiente manera:

**sec2ts** "PID del *Transport stream*" *entrada.sec salida.ts*

Por ejemplo:

**sec2ts** 0 pat.sec pat.ts

Crea un *Transport stream* llamado *pat.ts* con *PID 0* a partir de la sección *pat.sec*

Se procede a comentar la última parte del *script*:

#En primer lugar se creará el *Transport stream* correspondiente a la *NIT* con *PID 16* llamado *firstnit.ts*.

#Las 5 instrucciones siguientes crean un archivo llamado *nit.sec* donde se incluye sección de la *NIT* haciendo uso del método *pack* del objeto *nit*.

**out = open("./nit.sec", "wb")**

**out.write(nit.pack())**

**out.close**

**out = open("./nit.sec", "wb")**

**out.close**

#Se convierte la sección creada *nit.sec* en el *Transport stream* *firstnit.ts* con *PID 16*.

**os.system('/usr/local/bin/sec2ts 16 < ./nit.sec > ./firstnit.ts')**

#A continuación se creará el *Transport stream* correspondiente a la *PAT* con *PID 0* llamado *firstpat.ts*.

#Las 5 instrucciones siguientes crean un archivo llamado *pat.sec* donde se incluye sección de la *PAT*, haciendo uso del método *pack* del objeto *pat*.

**out = open("./pat.sec", "wb")**

**out.write(pat.pack())**

**out.close**

**out = open("./pat.sec", "wb")**

**out.close**

#Se convierte la sección creada *pat.sec* en el *Transport stream firstpat.ts* con *PID 0*.

**os.system('/usr/local/bin/sec2ts 0 < ./pat.sec > ./firstpat.ts')**

#A continuación se creará el *Transport stream* correspondiente a la *SDT* con *PID 17* llamado *firstsdt.ts*.

#Las 5 instrucciones siguientes crean un archivo llamado *sdt.sec* donde se incluye sección de la *SDT* haciendo uso del método *pack* del objeto *sdt*.

**out = open("./sdt.sec", "wb")**

**out.write(sdt.pack())**

**out.close**

**out = open("./sdt.sec", "wb")**

**out.close**

#Se convierte la sección creada *sdt.sec* en el *Transport stream firstsdt.ts* con *PID 17*.

**os.system('/usr/local/bin/sec2ts 17 < ./sdt.sec > ./firstsdt.ts')**

#A continuación se crea el *Transport stream* correspondiente a la *PMT* con el *PID* que se le haya asignado en la variable *avalpa\_pmt\_pid*, llamado *firstpmt.ts*.

#Las 5 instrucciones siguientes crean un archivo llamado *pmt.sec* donde se incluye sección de la *PMT* haciendo uso del método *pack* del objeto *pmt*.

**out = open("./pmt.sec", "wb")**

**out.write(pmt.pack())**

**out.close**

**out = open("./pmt.sec", "wb")**

**out.close**

#Se convierte la sección creada *pmt.sec* en el *Transport stream firstpmt.ts* con el *PID* indicado en *avalpa1\_pmt\_pid*.

```
os.system('/usr/local/bin/sec2ts ' + str(Avalpa1_PMT_pid) + ' < ./pmt.sec > ./firstpmt.ts')
```

#Se debe comentar que si se quisiera incluir otro servicio de televisión digital, se debería incluir otras 6 instrucciones que creasen un archivo *pmt2.sec* y que lo transformaran en otro *Transport stream*. Por ejemplo *firstpmt2.ts* a partir de su objeto *pmt2* y su *PID*, definidos anteriormente.

#A continuación se creará el *Transport stream* correspondiente a la *AIT* con el *PID* que hayamos puesto en la variable *ait1\_pid* llamado *firstait.ts*.

#Las 5 instrucciones siguientes crean un archivo llamado *ait.sec* donde se incluye la sección de la *AIT* haciendo uso del método *pack* del objeto *ait*.

```
out = open("./ait.sec", "wb")
```

```
out.write(ait.pack())
```

```
out.close
```

```
out = open("./ait.sec", "wb") # Python flush bug
```

```
out.close
```

#Se convierte la sección creada *ait.sec* en el *Transport stream* *firstait.ts* con el *PID* indicado en *ait1\_pid*.

```
os.system('/usr/local/bin/sec2ts ' + str(ait1_pid) + ' < ./ait.sec > ./firstait.ts')
```

#A continuación se creará el *Transport stream* correspondiente a la *EIT* con *PID 18* llamado *firsteit.ts*.

#Las 5 instrucciones siguientes crean un archivo llamado *eit.sec* donde se incluye la primera sección de la *EIT* haciendo uso del método *pack* del objeto *eit*.

```
out = open("./eit.sec", "wb")
```

```
out.write(eit.pack())
```

```
out.close
```

```
out = open("./eit.sec", "wb")
```

```
out.close
```

#Se convierte la sección creada *eit.sec* en el *Transport stream* *firsteit.ts* con el *PID* indicado en *ait1\_pid*.

```
os.system('/usr/local/bin/sec2ts 18 < ./eit.sec > ./firsteit.ts')
```



#A continuación se añade al *Transport stream* que incluye la *EIT* la segunda sección que contenía un evento: *eit\_follow*.

#Las 5 instrucciones siguientes crean un archivo llamado *eit\_follow.sec* donde se incluye la segunda sección de la *EIT* haciendo uso del método *pack* del objeto *eit\_follow*.

```
out = open("./eit_follow.sec", "wb")
```

```
out.write(eit_follow.pack())
```

```
out.close
```

```
out = open("./eit_follow.sec", "wb")
```

```
out.close
```

#Se añade al *Transport stream* *firsteit.ts* la parte de la sección *eit\_follow.sec*.

```
os.system('/usr/local/bin/sec2ts 18 < ./eit_follow.sec >> ./firsteit.ts')
```

#En este caso la segunda sección llamada *eit\_follow* debe ser incluida en el *Transport stream* del *EIT* *firsteit.ts* y no en un *Transport stream* nuevo como ocurre con los *Transport streams* correspondientes a las *PMT's* que necesitan un *Transport stream* por cada tabla *PMT* creada.

#Por último se creará el *Transport stream* correspondiente a la *TDI* con *PID* 20 llamado *firsttdt.ts*.

#Las 5 instrucciones siguientes crean un archivo llamado *tdt.sec* donde se incluye sección de la *TDI* haciendo uso del método *pack* del objeto *tdt*.

```
out = open("./tdt.sec", "wb")
```

```
out.write(tdt.pack())
```

```
out.close
```

```
out = open("./tdt.sec", "wb")
```

```
out.close
```

#Se convierte la sección creada *tdt.sec* en el *Transport stream* *firsttdt.ts* con *PID* 20.

```
os.system('/usr/local/bin/sec2ts 20 < ./tdt.sec > ./firsttdt.ts')
```

Aquí finaliza el script.

Se debe comentar que las secciones tienen una longitud máxima de bytes(1024 para las tablas PSI del estándar 13818-1 y 4096 para las secciones privadas) y que si alguna de las secciones es mayor que la longitud máxima o tiene algún error de sintaxis al ser definidos sus parámetros, el script no creará los *Transport streams* de las tablas.

Finalmente, si todos los parámetros han sido definidos de manera correcta al ejecutar el *script* se crearan, por un lado, todos los archivos de las secciones de las tablas: *nit.sec*, *pat.sec*, *sdt.sec*, *pmt.sec*, *ait.sec*, *eit.sec*, *tdt.sec*.

Por otro lado también se crearan los archivos de los *Transport streams* correspondientes a cada una de las tablas: *firstnit.ts*, *firstpat.ts*, *firstsdt.ts*, *firstpmt.ts*, *firstait.ts*, *firsteit.ts*, *firsttdt.ts*.

Estos archivos serán creados en el mismo directorio donde se encuentre el *script* en *Python*.

## 5.4 Comandos del Opencaster

### 5.4.1 Introducción

A continuación se verán ciertos aspectos de interés a la hora de usar las herramientas de *Opencaster*. También se explicarán los comandos más comunes usados en la emisión de los servicios de televisión digital, así como se indicará un listado completo de todos los comandos que posee *Opencaster*. Por último se expondrán varios ejemplos de uso de las herramientas a la hora de poner en emisión los servicios de televisión digital incluyendo los contenidos enviados de forma programada por el *VLC*.

### 5.4.2 Aspectos a tener en cuenta al usar los comandos de Opencaster

Todas las herramientas de las que dispone *Opencaster*, son comandos que se deben introducir a través del terminal de *Linux*. *Opencaster* no dispone de una interfaz gráfica que facilite el uso de sus herramientas. Su uso, en cambio, se puede simplificar si creamos un archivo de texto (*script*), al que deberemos dar permisos de ejecución, con los comandos que queramos utilizar, puestos en orden. Al ejecutar este *script* se ejecutarán todos los comandos en la *shell* de *Linux* (como si se ejecutaran en el *Terminal*) en el orden en el que fueron escritos.

Otro aspecto que se debe tener en cuenta es que el resultado de cada comando debe ser introducido en archivos distintos. Si usamos los comandos de *Opencaster* sobre archivos normales estos pueden llegar a tener un gran tamaño, así que es conveniente el uso de tuberías *fifo* para conectar las distintas operaciones efectuadas por los comandos. Las tuberías *fifo* (*first in-first out*) como su nombre indica se caracterizan por trabajar de forma que lo primero que entra es lo primero que sale. Estas no consumen espacio físico en el disco duro ya que solamente hacen de intermediarios entre procesos, es decir, enlazan la salida de un comando con la entrada de otro, lo que nos permite realizar tareas complejas en la línea de comandos. Así que, las tuberías *fifo* se quedan esperando a que otro proceso recoja la información. Así pues, la forma más habitual de utilizar los comandos será la siguiente:

*Comando parámetros tubería\_entrada.ts > tubería\_salida.ts*

Se deberán crear las tuberías previamente en el directorio de trabajo haciendo uso del comando *Linux*:

#### **mkfifo tubería.ts**

Además, interesa que los comandos se ejecuten en *background*. Para poder hacer esto se coloca el símbolo *&* al final de la línea del comando. De esta manera el comando se ejecutará y será posible utilizar el mismo terminal de *Linux* para ejecutar otros comandos.

### 5.4.3 Descripción de los comandos *Opencaster* más comunes a la hora de emitir servicios de televisión digital.

#### 5.4.3.1 *oc-update.sh*.

Comando *Opencaster* para la creación de un *Transport stream* con un carrusel de objetos *DSMCC MHP* a partir de un directorio que contenga las clases de la aplicación. Crea un *Transport stream* con una aplicación *MHP* dentro de ella.

Para poder crear el *Transport stream* se deben tener todas las clases de la aplicación *MHP* en un directorio. Este directorio será el directorio raíz para el carrusel de objetos. Al ejecutar el comando se creará un *Transport stream* con el mismo nombre que tenga el directorio donde se alojan las clases de la aplicación.

El comando se usa de la siguiente manera:

***oc-update.sh directorio\_aplicacion association\_tag version PID  
carousel\_id [compresion\_habilitada] [relleno\_habilitado]  
[borrado\_deshabilitado]***

Por ejemplo:

*oc-update.sh aplicacion 0xB 5 2003 1 1 0 0*

- *directorio\_aplicacion*: será el directorio que contiene las clases de la aplicación *MHP*.
- *0xB*: es la *association\_tag* indicada en las tablas *AIT* y *PMT*.
- *5*: es el número de versión del carrusel. Si se actualiza la aplicación se deberá cambiar a 6 para que el decodificador se de cuenta.
- *2003*: Es el numero del *PID* del *Transport stream* que se crea.
- *1* es la ID del carrusel indicada en la *PMT*.
- *1* hace la compresión del carrusel.
- *0* no hace relleno.
- *0* borra los archivos temporales una vez ha terminado de crear el *Transport stream*.

El ejemplo dará lugar a un *Transport stream* con *PID 2003* llamado *directorio\_aplicacion.ts* con un carrusel de objetos *MHP* cuyo *asociation\_tag* es *0xB* y su *carousel\_id* es *1*.

#### 5.4.3.2 *tsudpreceive*

*Opencaster* posee una herramienta que será útil a la hora de obtener los distintos vídeos que queramos emitir para su posterior multiplexación con las tablas creadas y las aplicaciones interactivas. Esta herramienta abre un *socket UDP* compuesto de una dirección *multicast* y un puerto y queda a la espera de recibir un *Transport stream*. Su sintaxis es la siguiente.

***tsudpreceive direccion\_multicast puerto > salida.ts***

Como se verá más adelante este comando será útil para recibir los contenidos que enviará el VLC de manera programada a través de una red.

Veamos un ejemplo:

### **tsudpreceive 224.0.0.1 1234 > Transport\_stream\_recibido.ts**

Para poder usar este comando el sistema *Linux* debe tener configurado el direccionamiento *multicast*. Si no lo estuviera se debe configurar una ruta, para ello simplemente se debe introducir el siguiente comando en el terminal de *Linux*.

**sudo route add -net 224.0.0.1 netmask 240.0.0.0 dev eth0**

### **5.4.3.3 tsfilter**

Este comando *Opencaster* se usa para el filtrado de *PID's*.

*tsfilter* sirve para extraer los paquetes correspondientes a ciertos *PID's* de un *Transport stream* desechando el resto. Esto puede ser útil cuando se pretenda extraer de un *Transport stream* multiplexado con las tablas *PSI-SI* sólo sus paquetes de audio y de vídeo.

El comando se usa de la siguiente forma:

**tsfilter entrada.ts +pid1 +pid2... > entrada\_filtrada.ts**

*entrada\_filtrada.ts* será un *Transport stream* que contendrá sólo los *PID's* que le indiquemos.

Un ejemplo de su utilización:

**tsfilter Transport\_stream\_recibido +2064 +2068 > Transport\_stream\_filtrado.ts**

*Transport\_stream\_filtrado* es un *Transport stream* que contiene solamente los paquetes que posean *PID 2064* y *2068*.

### **5.4.3.4 tspidmapper**

Comando *Opencaster* para re-mapeo de *PID's*.

Con *Opencaster* se puede asignar a los paquetes de un *Transport stream* con un cierto *PID*, otro *PID* diferente. Esto es útil cuando se pretende cambiar a un *Transport stream* de audio o vídeo su *PID* original por el *PID* que se haya especificado en la sección de la tabla *PMT* del *script* en *Python*.

Su uso es sencillo:

**tspidmapper entrada.ts pid1 to pid1nueva and pid2 to pid2 nueva... > entrada\_remapeada.ts**

Cambia el *PID* de los paquetes que tengan como *PID pid1* y los pone como *PID pid1nueva*. Este comando será especialmente útil a la hora de tener 2 servicios.

Veamos un ejemplo:

***tspidmapper Transport\_stream\_filtrado.ts 2064 to 2071 and 2068 to 2072 > Transport\_stream\_remapeado.ts***

Se cambia el *PID* de los paquetes que tengan *PID 2064* a *2071* y los paquetes que tengan *PID 2068* a *2072* de *Transport\_stream\_filtrado.ts* y el resultado se guarda en *Transport\_stream\_remapeado.ts*

#### 5.4.3.5 tscbrmuxer.

Para poder multiplexar varios *Transport stream* en un solo *Transport stream* a tasa de bit constante *Opencaster* posee el comando *tscbrmuxer*. Su forma de uso es la siguiente:

***tscbrmuxer b:4000000 transportstream1.ts b:4000000 transportstream2.ts > transportstreammultiplexado.ts***

Se observa que para cada *Transport stream* debe ser especificada la tasa de bit (*bitrate*) que debe tener en bits por segundo. El *Transport stream* multiplexado tendrá una tasa de bit suma de las tasas de cada uno.

Veamos un ejemplo del multiplexado de un *Transport stream* de audio, con uno de vídeo, con los *Transport stream* de las tablas que creamos con el *script* ejemplo más un *Transport stream* que aloja una aplicación interactiva.

***tscbrmuxer b:4000000 vídeo.ts b:128000 audio.ts b:1000000 aplicacion.ts b:3008 firstpat.ts b:3008 firstpmt.ts b:1500 firstsdt.ts b:1400 firstnit.ts b:2000 firsteit.ts b:2000 firsttdt.ts b:2000 firstait.ts > Transport\_stream\_multiplexado.ts***

En este ejemplo se ha supuesto que se tiene un *Transport stream* de vídeo, codificado a tasa de bit constante de valor 4000000 bits por segundo y un *Transport stream* de audio asociado al vídeo, codificado también a tasa de bit constante de valor 128000 bits por segundo. Se debe hacer notar que debido a que el audio y vídeo van en 2 *Transport stream* diferentes sus tasas de bits deben ser las adecuadas para su correcta sincronización. Este hecho no es trivial ya que se necesita de una codificación previa adecuada y un conocimiento exacto de la tasa de bit de los contenidos codificados. Una pequeña variación en la sincronización del audio y vídeo pueden dar lugar a una emisión deficiente.

El problema anterior se puede simplificar si se logra que, en vez de tener el audio y vídeo en *Transport streams* separados, se tenga un único *Transport stream* que aloje tanto el audio como el vídeo de forma que ya estén sincronizados y para el cual se sepa su tasa de bits exacta y constante.

Se debe comentar también los valores de las tasas de bit de las tablas. Estos valores no son arbitrarios y vienen fijados por el tiempo máximo que puede pasar entre la recepción de cada tabla. La *PAT* y la *PMT*, por ejemplo, deben ser emitidas cada 500 ms. Dependiendo del tamaño de *firstpat.ts* se deberá calcular la tasa de bit. Si *firstpat.ts* tiene un tamaño de 188 bytes (1 paquete *TS*) se deberán mandar 188 bytes \* 8 bits=1504 bits cada 500ms es decir 1504bits/0.5sg=3008 bits por segundo. El cálculo con el resto de las tablas se hace de manera parecida.

Por ultimo, se debe tener en cuenta que para que el decodificador vea el vídeo, el audio y reconozca la aplicación que se ha multiplexado en el ejemplo, sus *Transport streams* deben tener como *PID's*, los *PID's* indicados en las tablas *PMT* y *AIT*; en este caso 2064 para el *Transport stream* de vídeo 2068 para el *Transport stream* de audio y 2003 para el *Transport stream* de la aplicación.

#### 5.4.3.6 tstdt

El comando *tstdt* permite cambiar de forma automática la tabla *tdt* de un *Transport stream* de manera sencilla. Modifica los valores de la fecha y hora cambiándolos por la fecha y hora actual del ordenador. Esto es esencial para que los eventos del *EGP* se lancen a la hora adecuada.

El comando se usa de la siguiente manera:

***tstdt entrada.ts > salida.ts***

Reemplaza los paquetes correspondientes a la tabla *tdt* que de *entrada.ts* por paquetes *tdt* con la hora actual.

Un ejemplo de la utilización del comando:

***tstdt Transport\_stream\_multiplexado.ts > Transport\_stream\_con\_fecha\_y\_hora\_actualizada.ts***

#### 5.4.3.7 Tsstamp, tspcrstamp

*Opencaster* también posee dos comandos que se ocupan de arreglar las marcas de referencia de reloj (*PCR*) de un *Transport stream* así como las marcas de decodificación (*DTS*) y presentación (*PTS*) de un *Transport stream* de vídeo susceptibles de ser incorrectas después de la multiplexación con el comando *tsctxmuxer*. Tanto las marcas de referencia de reloj, *PCR*, las marcas de decodificación, *DTS*, y las marcas de presentación, *PTS*, son muy importantes para que el decodificador pueda presentar los contenidos de manera correcta en el televisor. Las marcas de referencia de reloj *PCR* son propias de los *Transport streams* [*Pcr1*] mientras que las marcas de decodificación *DTS* y las de presentación *PTS* [*Pts1*] son propias de los *PES* [Header *PES*].

Sus sintaxis son las siguientes:



***tsstamp entrada.ts tasadebits > salida.ts***

***tspcrstamp entrada.ts tasadebits > salida.ts***

Son muy parecidos entre si, la diferencia entre ellos es que *tsstamp* modifica el *PCR/DTS/PTS* y *tspcrstamp* sólo modifica el *PCR*. Para el valor de *tasadebits* debe ser especificado la tasa de bits que tiene el *Transport stream* de entrada en bits por segundo.

Dos ejemplos del uso de estos comandos:

***tsstamp Transport\_stream\_multiplexado.ts 19906000 > Transport\_stream\_corregido\_DTS\_PTS\_PCR.ts***

***tspcrstamp Transport\_stream\_multiplexado.ts 19906000 > Transport\_stream\_corregido\_PCR.ts***

Estos dos comandos realizarán los cambios oportunos en las cabeceras del *Transport stream* así como en las cabeceras de los *PES* incluidos en ellas para corregir de manera correcta los valores de los *PCR/DTS/PTS* adaptándolos a la tasa de bit introducida como parámetro y los guardarán en *Transport\_stream\_corregido.ts*.

#### **5.4.4 Listado de todas las herramientas disponibles en Opencaster**

*Dsmcc-receive*: Comando que extrae un carrusel de objetos *DSMCC* de un *Transport stream*.

*Esaudio2pes*: Comando que encapsula un *elementary stream* de audio en un *paquetized elementary stream*.

*Esaudioinfo*: Comando que analiza y muestra en la pantalla la información de un *elementary stream* de audio.

*Esvideo2pes*: Comando que encapsula un *elementary stream* de vídeo en un *paquetized elementary stream*.

*I13942ts*: Comando que transforma un archivo volcado a través de una conexión *firewire* (iee 1394 de 6 pines) en un *Transport stream*.

*Pes2es*: Comando que desencapsula un *Paquetized elementary stream* en un *Elementary stream* ya bien sea de audio o de vídeo.

*Pesaudio2ts*: Comando que transforma un *Paquetized elementary stream* de audio en un *Transport stream*.

*Pesinfo*: Comando que analiza y muestra por pantalla la información de un *Paquetized elementary stream*.

*Pesvideo2ts*: Comando que transforma un *Paquetized elementary stream* de

*vídeo en un Transport stream.*

*Sec2ts:* Comando que transforma una sección en un *Transport stream*.

*Ts2pes:* Comando que transforma un *Transport stream* en un *Paquetized elementary stream*.

*Tscbrmuxer:* Comando que multiplexa varios *Transport streams* en uno sólo.

*Tsdoubleoutput:* Comando que duplica un *Transport stream*.

*Tsfilter:* Comando que hace un filtrado de *PID's* en un *Transport stream*.

*Tsfixcc:* Comando que corrige el *Continuity Counter* de un *Transport stream*.

*Tsinputswitch:* Comando que conmuta su *Transport stream* de salida entre dos *Transport streams* de entrada.

*Tsloop:* Comando que hace un lazo sin fin de un *Transport stream*.

*Tsmodder:* Comando que reemplaza los paquetes de un *Transport stream* con un cierto *PID* encapsulado en un *Transport stream* multiplexado, por otro *Transport stream* indicado como entrada.

*Tsnullfiller:* Comando que añade paquetes vacíos a un *Transport stream* para alcanzar una tasa de bits.

*Tsnullshaper:* Comando que reemplaza paquetes vacíos con paquetes de un *Transport stream*.

*Tsoutputswitch:* Comando que conmuta su entrada entre 2 *Transport stream* de salida o en ambos.

*Tspcrmeasure:* Comando que analiza los *PCR's* de un *Transport stream* y estima a partir de ellos la tasa de bit de un *Transport stream*.

*Tspcrstamp:* Comando que corrige los *PCR's* de un *Transport stream*.

*Tspidmapper:* Comando que cambia los *PID's* de los *Transport streams* de un *Transport stream* multiplexado por otros *PID's*.

*Tsstamp:* Comando que corrige los *PCR's*, los *PTS's* y los *DTS's* de un *Transport stream*.

*Tstcpreceive:* Comando que recibe un *Transport stream* a través de un *socket TCP*.

*Tstcpsend:* Comando que envía un *Transport stream* a través de un *socket TCP*.

*Tstdt*: Comando que cambia la fecha y hora de la tabla *TDT* de un *Transport stream* por la fecha y hora actual del ordenador.

*Tstimedwrite*: Comando que escribe un *Transport stream* de entrada en un *Transport stream* de salida a una cierta tasa de bits.

*Tstimeout*: Comando que permite hacer una pausa en un *Transport stream*.

*Tsudpreceive*: Comando que recibe un *Transport stream* a través de un *socket UDP* de una dirección *multicast*.

*Tsudpsend*: Comando que recibe un *Transport stream* a través de un *socket UDP* de una dirección *multicast*.

*Tsvbr2cbr*: Comando que permite transformar un *Transport stream* que contenga un *vídeo* con tasa de bits variable en un *Transport stream* de tasa de bit constante añadiendo paquetes vacíos.

*Vbv*: Comando que se utiliza para simular un *buffer* de un decodificador de *vídeo*.

## 5.5 Emisión de un *Transport stream* con la tarjeta Dektec

*DtPlay* no es un comando propio de *Opencaster*. Es un comando para operar con la tarjeta moduladora *Dektec*.

Para proceder a la modulación de un *Transport stream* con la tarjeta *Dektec* para su emisión "al aire" se debe hacer uso del comando *DtPlay*. El comando *DtPlay* funciona indicándole el archivo que se va a poner en emisión y definiendo los parámetros de modulación. Por ejemplo:

```
DtPlay archivo.ts -t 110 -mt OFDM -mC QAM64 -mG 1/4 -mc 2/3 -mf 578 -  
mB 8 -mT 8
```

El comando pondrá en emisión a través de la tarjeta *Dektec* el *Transport stream* "archivo.ts" con los siguientes parámetros:

- t 110: le indica que va a usar la tarjeta *Dektec* 110T.
- mt OFDM: le indica que vamos a usar la modulación *OFDM*.
- mC QAM64: indica el tipo de constelación de DVB que se usará.
- mG 1/4: el intervalo de guarda.
- mc 2/3: la tasa del código convolucional.
- mf 578: la frecuencia central donde se modula en megahercios.
- mB 8: indica el ancho de banda en megahercios.
- mT 8: es el modo de transmisión de la *FFT*.

Todos estos parámetros son propios de las emisiones del estándar *DVB* que utiliza la modulación *COFDM*. Será importante saber qué parámetros de modulación *COFDM* estamos usando ya que éstos definen la tasa de bit que debe tener el *Transport stream* a emitir.

Uso de las herramientas de Opencaster y VLC para la emisión de contenidos de manera programada.

El objetivo principal propuesto es poder emitir vídeos de manera programada y que, con la ayuda de las herramientas que *Opencaster* nos proporciona, las pongamos en emisión. Para poder servir los vídeos se usará el *VLC* que se ocupará de mandar los vídeos a través de la red. Se tendrá un *socket UDP* escuchando, creado con el comando *Opencaster tsudpreceive*.

El programa que se utilizara es el *VLC*, un programa de libre distribución capaz de mandar los vídeos a la dirección *multicast* que se desee. Además, este programa permite adaptar los vídeos para su uso con las herramientas de *Opencaster* ya que es capaz de re-codificarlos a *Transport streams* con las propiedades que nosotros queramos, por ejemplo, la tasa de bit de audio y de vídeo, asignarles los *PID's* al audio y al vídeo, etc.

### 5.5.1 Pasos previos a la puesta en emisión

En primer lugar se deberá crear un directorio de trabajo. Todos los elementos que se utilicen (*script* en *shell* de *Linux*, *script* en *Python*, tuberías *Transport streams* de las tablas, *Transport stream* de las aplicaciones, *Transport stream* de paquetes vacíos, directorio con las clases de las aplicaciones) deberán estar situados en dicho directorio de trabajo. Además, todos los comandos deben ser ejecutados en el directorio de trabajo.

Una vez modificado el *script* en *Python*, descrito en la *sección 3.2* y situado en el directorio de trabajo, deberá ser ejecutado. De esta manera serán creados los *Transport streams* de las tablas en el directorio de trabajo:

- *firstpat.ts*
- *firstpmt.ts*
- *firstsdt.ts*
- *firstnit.ts*
- *firsteit.ts*
- *firstait.ts*
- *firsttdt.ts*

También se deberá ejecutar el comando que crea el *Transport stream* de la aplicación a partir de una carpeta que contiene las clases de las aplicaciones. Por ejemplo en una carpeta llamada *mhp*.

**oc-update.sh mhp 0xB 5 2003 1 1 0 0**

Tras ejecutar el comando se dispondrá de un *Transport stream* con el carrusel de objetos que contiene las clases de las aplicaciones llamado:

- *mhp.ts*

Es necesario también tener en el directorio de trabajo el *Transport stream* de los paquetes vacíos así que se deberá copiar un archivo llamado *null.ts* que es un *Transport stream* con *PID 8191* que contiene paquetes vacíos. Este archivo nos lo proporciona *Opencaster*.

Todos los comandos, deben trabajar con tuberías *fifo*, así que se deberán crear las tuberías previamente a la puesta en emisión.

Se deberán crear todas las tuberías que se vayan a utilizar. Esto sólo se tendrá que hacer la primera vez que se vaya a poner el servicio en emisión. Para la emisión de un servicio deberemos crear las siguientes tuberías

**mkfifo      vlan.ts      vlanfilter.ts      playout.ts      playouttdt.ts  
playoutstamped.ts**

### 5.5.2 Script de comandos Opencaster para la emisión de un servicio de televisión digital

A continuación se comentará un *script* con comandos *Opencaster* que hacen posible la emisión de un servicio de televisión digital. El servicio contará con información de la EPG así como de aplicaciones *MHP*.

El *script* se puede ejecutar, si se le conceden permisos de ejecución al mismo. Otra posibilidad es copiar los comandos del *script* en el terminal.

- **Primer paso: Recepción de los contenidos**

Se crea el *socket UDP* en la dirección 224.0.0.1 y el puerto 1234 y se guardamos en la tubería *vlan.ts*:

**tsudpreceive 224.0.0.1 1234 > vlan.ts &**

- **Segundo paso: Filtrado de audio y vídeo**

El *VLC* manda un *Transport stream* que posee sus propias tablas *PAT* y *PMT*. El objetivo es extraer sólo el audio y vídeo que el *VLC* manda para mas adelante multiplexarlo con los *Transport stream* de las tablas *PSI*. Para ello se filtrará el *Transport stream* recibido con el comando *tsfilter*, manteniendo solo el audio y vídeo.

**tsfilter vlan.ts +2064 +2068 > vlanfilter.ts &**

EL *VLC* permite configurar las *PID's* de audio y vídeo. En este ejemplo se configuró, con el *VLC*, como *PID* de Audio la 2068 y *PID* de Vídeo la 2064. Son los mismos valores que se definieron en la *PMT* con el *script* en *Python*. Si se configuraran con el *VLC* otros valores para las *PID's* de audio y vídeo, únicamente se debería usar el comando *tspidmapper* para cambiarles las *PID's* por las que fueron definidas en la *PMT*.

- **Tercer paso: Multiplexacion del Transport stream**

El siguiente comando, *tscbrmuxer* se ocupa de multiplexar *vlanfilter.ts*, donde se encuentran el audio y vídeo multiplexados en un único *Transport stream* (con lo que nos ahorramos problemas de sincronismo de audio y vídeo), con los *Transport stream* de las tablas creadas con el *script* en *Python* y el *Transport stream* de las aplicaciones.

A la hora de usar *tscbrmuxer* se debe tener en cuenta que la tasa de bits total que debe tener el *Transport stream* multiplexado depende de los parámetros de modulación.

En este ejemplo se usa una modulación *QAM64* con intervalo de guarda de 1/4 y con un la tasa de bit de código convolucional de 2/3 lo que requiere una tasa de bit del *Transport stream* multiplexado de 19906000 bits/segundo.

Además, se debe conocer la tasa de bit del *Transport stream* del audio y vídeo multiplexado, así como las tasas de bit mínimas necesarias para cada una de las tablas y el carrusel de objetos de la aplicación. Como la suma de la tasa de bits del audio y vídeo multiplexados, las tablas y la aplicación no llega a ser la requerida por la modulación se debe rellenar el *Transport stream* multiplexado con paquetes sin información contenidos en el *Transport stream null.ts*, que tiene *PID 8191*, hasta alcanzar la tasa requerida por los parámetros de modulación.

Se pueden ver las distintas tasas de bit disponibles dependiendo de los parámetros de modulación [Tabla tasa de bits].

Para que, cuando se envíen los contenidos con el *VLC*, la tasa de bit de audio y vídeo multiplexados sea siempre la misma, se deberá hacer un procesamiento previo de los contenidos. Para ello se debe recodificar todos los contenidos y encapsularlos en *Transport stream* con la ayuda de un *encoder* de la empresa *Mainconcept*; el *Mainconcept Reference*. La forma de codificar con el *Reference* de *Mainconcept* se explicará más adelante.

Una vez re-codificados se deberá hacer una prueba experimental para saber la tasa de bit que se debe introducir al comando *tscbrmuxer*. Se pretende averiguar cual es la tasa de bit para el *Transport stream* que sólo contiene el audio y vídeo es decir la tasa de bit de *vlanfilter.ts*.

Esta prueba es necesaria ya que al transformar los *Elementary streams* de audio y vídeo, en *Transport stream*, la tasa de bit del *Transport stream* es mayor a la suma de las tasas de bit de los *Elementary streams*. Esto se debe a que los paquetes de *Transport stream* tienen un tamaño de 188 bytes de los cuales 4 bytes suelen ser de cabecera. Se le añade, de esta manera una carga debido a las cabeceras de los paquetes de *Transport stream*. Esta carga causada por las cabeceras suele ser de media del orden de un 3.63% más.

Por otro lado, al transformar a *Transport stream*, el *Reference* de *Mainconcept* añade la información de las tablas *PAT* y *PMT* lo cual hace aumentar también su tasa de bit. A continuación, el *VLC* cambia la



información de las tablas *PAT* y *PMT*. Si se analiza la salida del *VLC* directamente, para saber la tasa de bit, no se obtendría el valor que se busca ya que a la salida del *VLC*, se le filtran todas las *PID*'s menos las de video y audio. El *Transport stream* que realmente se multiplexa con el *tsmuxer* es el ya filtrado, y es de este precisamente del que se quiere saber su tasa de bit exactamente.

Esta prueba permite saber el valor de tasa de bit que se debe introducir al *tsmuxer* para unos parámetros de codificación de vídeo y audio introducidos en el *Reference* de *Mainconcept*.

Para realizar la prueba se deberá capturar y analizar *vlanfilter.ts*. Para ello se deben ejecutar los dos comandos anteriores pero en el segundo comando en vez de mandar la salida a una tubería se debe poner simplemente un nombre de archivo que no sea una tubería. Por ejemplo:

**tsudpreceive 224.0.0.1 1234 > vlan.ts &**

**tsfilter vlan.ts +2064 +2068 > prueba.ts &**

*prueba.ts* no es una tubería. Una vez lanzados estos dos comandos, en *prueba.ts* se empezará a guardar el *Transport stream* del audio y vídeo sin sus tablas. Una vez pasado un tiempo se debe matar estos dos procesos ya que *prueba.ts* puede llegar a ser demasiado grande y llenar completamente el disco. Para matar los procesos se introduce en el *Terminal*, en primer lugar:

**ps -e**

Este comando muestra una lista de todos los procesos que se están ejecutando y sus identificadores. Es necesario conocer el identificador del proceso para matarlo. A continuación se matan los procesos con el comando *kill*:

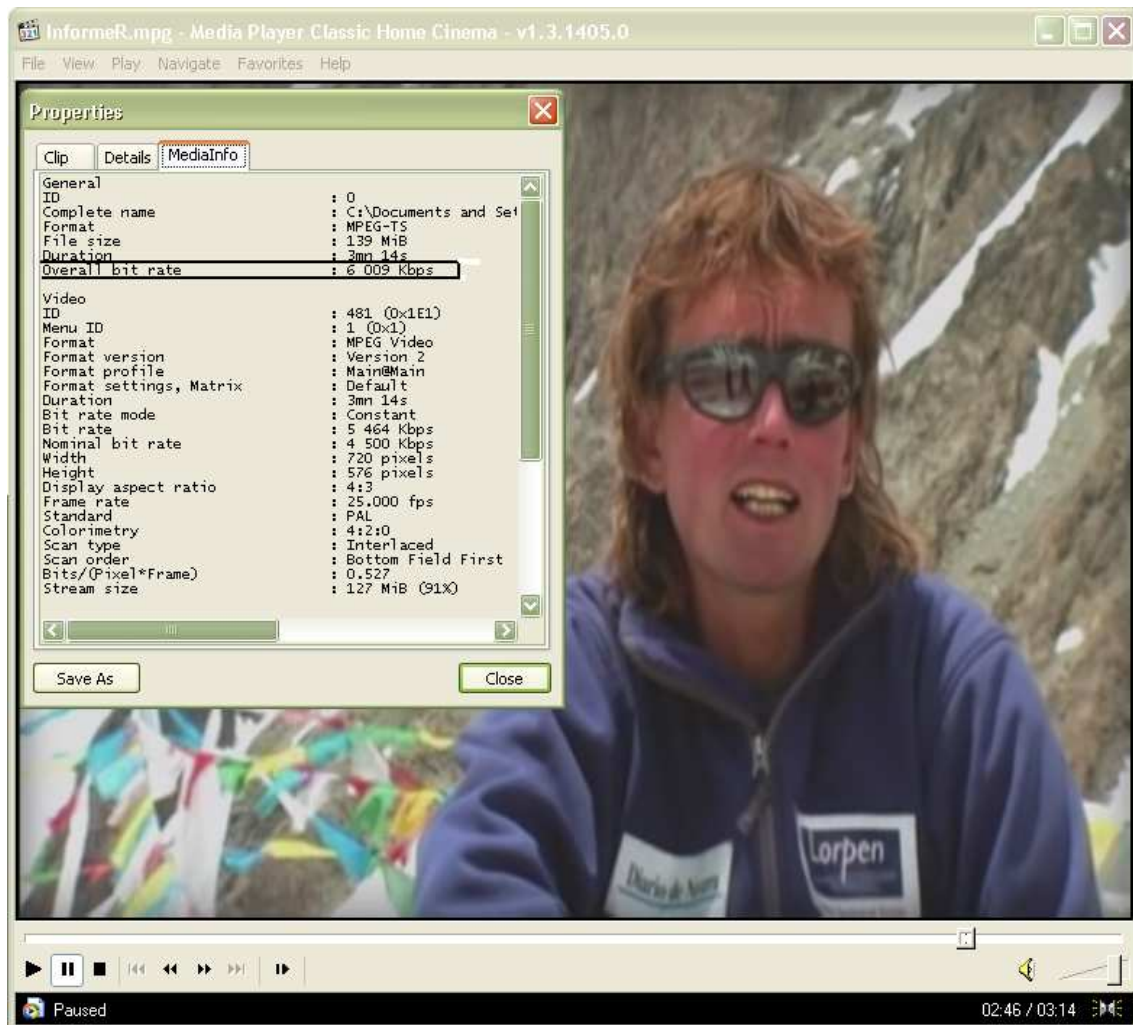
**kill id\_de\_tsudpreceive id\_de\_tsfilter**

De esta manera se tendrá en *prueba.ts* un *Transport stream* que contiene solamente audio y vídeo.

Por ultimo, se debe analizar *prueba.ts* para saber su tasa de bit. Para ello se hace uso de un reproductor multimedia que da una información muy detallada de los parámetros de los archivos que reproduce. Este programa es el *Media Player Classic Home Cinema* y es de libre distribución. Se puede bajar en [Mpc1].

Para analizarlo se abre *prueba.ts* con el *Media Player Classic* y pulsando *Shift+F10* accedemos a las propiedades del archivo. También se puede acceder a las propiedades pinchando con el ratón en *File* y eligiendo en el menú desplegable *properties*. En las propiedades se debe pulsar la pestaña *Media Info*. Para saber el valor de la tasa de bit debemos observar el valor que indica *Overall bit rate* como se puede observar en la *figura 31*.

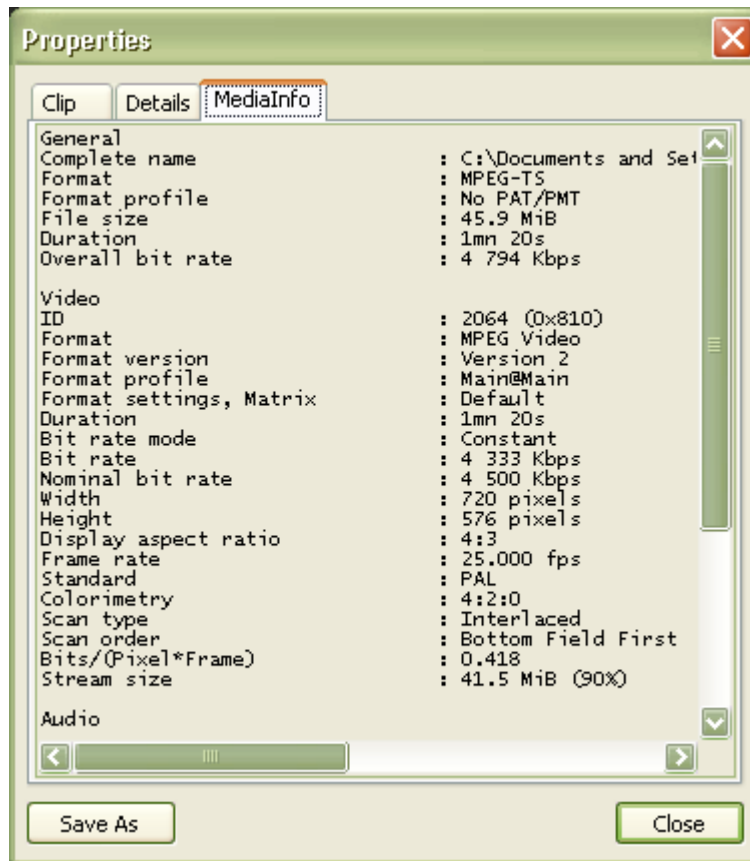




**Figura 31**

Ese valor será redondeado al alza a un múltiplo de  $188\text{bytes} \times 8\text{bits} = 1508$  y este será el valor que se debe introducir a *tscbrmuxer*.

Por ejemplo, para todos los contenidos cuyos vídeos han sido codificados a 4500000 bits por segundo y su audio a 128000 con el *Reference* de *Mainconcept* el valor obtenido de tasa de bit de *vlanfilter.ts* es de 4795504.



**Figura 32**

Las tasas de bit necesarias para las tablas dependen de la frecuencia con la que deben ser mostradas y la longitud de la tabla en paquetes de *Transport stream* que tienen. Un paquete de *Transport stream* tiene una longitud de 188 bytes = 1504 bits. Muchas veces las tablas están contenidas en uno o dos paquetes así que no suelen ser mayores a 3008 bits. Las tablas más grandes suelen ser las de la información de eventos es decir las tablas de la *EIT*. A continuación mostraremos las tasas de bits mínimas necesarias para tablas que van contenidas en 1 paquete.

- PAT 3008 bits/segundo
- PMT 3008 bits/segundo
- SDT 1500 bits/segundo
- NIT 1400 bits/segundo
- EIT 2000 bits/segundo
- TDT 2000 bits/segundo
- AIT 2000 bits/segundo

Estas tasas de bit son las mínimas pero pueden ser mayores. Para no estar en el límite se usarán tasas de bit para todas las tablas de 10000 bits por segundo.

Una vez comentado todo esto se muestre el comando que multiplexa el *Transport stream*.

**tscbrmuxer b:4795504 vlanfilter.ts b:1000000 mhp.ts b:10000**

**firstait.ts b:10000 firstpat.ts b:10000 firstpmt.ts b:10000 firstsdt.ts  
b:10000 firstnit.ts b:10000 firsteit.ts b:10000 firsttdt.ts b:14040496  
null.ts > playout.ts &**

- **Cuarto paso: Corrección de la hora**

Como se pretende que el *Transport stream* lleve la hora actual se actúa sobre el *Transport stream* con el comando *tstdt* para ese propósito. Este comando recoge la hora directamente del ordenador, así que es importante que el ordenador este siempre en hora. Para esto sería importante, que el ordenador estuviera conectado a un servidor que le actualizara la hora periódicamente.

**tstdt playout.ts > playouttdt.ts &**

- **Quinto paso: Corrección de las marcas de referencia de reloj**

Se corrigen las marcas de referencia de reloj con el comando *tspcrstamp*.

**tspcrstamp playouttdt.ts 19906000 > playoutstamped.ts &**

- **Sexto y ultimo paso: Emision**

Finalmente se pone en emisión el *Transport stream* multiplexado *playoutstamped.ts*.

**DtPlay playoutstamped.ts -t 110 -mt OFDM -mC QAM64 -mG 1/4 -  
mc 2/3 -mf 578**

El *script* completo queda simplemente así:

**tsudpreceive 224.0.0.1 1234 > vlan.ts &**

**tsfilter vlan.ts +2064 +2068 > vlanfilter.ts &**

**tsnbrmuxer b:4795504 vlanfilter.ts b:1000000 mhp.ts b:10000  
firstait.ts b:10000 firstpat.ts b:10000 firstpmt.ts b:10000 firstsdt.ts  
b:10000 firstnit.ts b:10000 firsteit.ts b:10000 firsttdt.ts b:14040496  
null.ts > playout.ts &**

**tstdt playout.ts > playouttdt.ts &**

**tsstamp playouttdt.ts 19906000 > playoutstamped.ts &**

**DtPlay playoutstamped.ts -t 110 -mt OFDM -mC QAM64 -mG 1/4 -  
mc 2/3 -mf 578**

### 5.5.3 *Script* de comandos Opencaster para la emisión de dos servicios de televisión digital

A continuación se comentará un *script* que tiene todos los comandos necesarios para la emisión de dos servicios de televisión digital. Un multiplex puede contener hasta cuatro servicios para estar completamente utilizado. Estas van a ser las pautas para poner dos servicios en él, pero el proceso es igual, salvo algunos parámetros, si se quiere incluir tres o hasta los cuatro servicios.

#### 5.5.3.1 Modificación del *script* en *Python*

Para la emisión de dos servicios de televisión digital se deberá modificar adecuadamente el *script* en *Python* para la creación de las tablas *PSI* de dos servicios, añadir todos los parámetros necesarios y crear el objeto correspondiente a la sección del *PMT* del nuevo servicio de televisión digital.

Se comentarán las partes a añadir en el *script*:

En la parte de variables se deberá añadir el identificador del servicio y la *PID* de la *PMT* del servicio:

```
avalpa1_service_id2 = 2
```

```
avalpa1_pmt_pid2 = 1033
```

#Al objeto *nit* se le añade el segundo servicio:

```
nit = network_information_section(  
    network_id = 1,  
    network_descriptor_loop = [  
        network_descriptor(network_name = "Avalpa",),  
    ],  
    transport_stream_loop = [  
        transport_stream_loop_item(  
            transport_stream_id = avalpa_transport_stream_id,  
            original_network_id =  
            avalpa_original_transport_stream_id,  
            transport_descriptor_loop = [  
                service_list_descriptor(
```

```
        dvb_service_descriptor_loop = [  
            service_descriptor_loop_item(  
                service_ID = avalpa1_service_id,  
                service_type = 1,  
            ),  
#Se añade el segundo servicio.  
            service_descriptor_loop_item(  
                service_ID = avalpa1_service_id2,  
                service_type = 1  
            ),  
        ],  
    ),  
],  
),  
],  
version_number = 1,  
section_number = 0,  
last_section_number = 0,  
)  
  
#En el objeto de la PAT se añade el segundo servicio y se define el PID de  
su PMT.  
  
pat = program_association_section(  
    transport_stream_id = avalpa_transport_stream_id,  
    program_loop = [  
        program_loop_item(  
            program_number = avalpa1_service_id,
```

**PID = avalpa1\_pmt\_pid,**

**),**

#Se le añade el servicio y se define el *PID* de su *PMT*.

**program\_loop\_item(**

**program\_number = avalpa1\_service\_id2,**

**PID = avalpa1\_pmt\_pid2,**

**),**

**program\_loop\_item(**

**program\_number = 0,**

**PID = 16,**

**),**

**],**

**version\_number = 1,**

**section\_number = 0,**

**last\_section\_number = 0,**

**)**

#En el objeto de la *SDT* se añade también el servicio y se definen sus propiedades.

**sdt = service\_description\_section(**

**transport\_stream\_id = avalpa\_transport\_stream\_id,**

**original\_network\_id = avalpa\_original\_transport\_stream\_id,**

**service\_loop = [**

**service\_loop\_item(**

**service\_ID = avalpa1\_service\_id,**

**EIT\_schedule\_flag = 0,**

**EIT\_present\_following\_flag = 0,**

```
        running_status = 4,  
  
        free_CA_mode = 0,  
  
        service_descriptor_loop = [  
            service_descriptor(  
                service_type = 1, # digital television service  
                service_provider_name = "UPNA",  
                service_name = "Upna TV",  
            ),  
        ],  
    ),
```

#Se añade el segundo servicio y se definen sus propiedades

```
    service_loop_item(  
        service_ID = avalpa1_service_id2,  
  
        EIT_schedule_flag = 0,  
  
        EIT_present_following_flag = 0,  
  
        running_status = 4,  
  
        free_CA_mode = 0,  
  
        service_descriptor_loop = [  
            service_descriptor(  
                service_type = 1, # digital television service  
                service_provider_name = "UPNA",  
                service_name = "Upna TV2",  
            ),  
        ],  
    ),
```

```
],  
  
version_number = 1,  
  
section_number = 0,  
  
last_section_number = 0,  
  
)
```

#Se crea un objeto llamado *pmt2* de la clase *program\_map\_section* para definir la *PMT* del segundo servicio: Los *PID*'s de vídeo y audio serán 2070 y 2071 respectivamente y la información de los *PCR*'s va incluida en el *PID* 2070, es decir en el vídeo.

```
pmt2 = program_map_section(
```

#Se define a qué servicio corresponde la *PMT* y el *PID* de los *PCR*'s.

```
    program_number = avalpa1_service_id2,  
  
    PCR_PID = 2070,  
  
    program_info_descriptor_loop = [],  
  
    stream_loop = [  

```

#Se definen los *PID*'s de vídeo y audio .

```
        stream_loop_item(  
  
            stream_type = 2,  
  
            elementary_PID = 2070,  
  
            element_info_descriptor_loop = []  
  
        ),  
  
        stream_loop_item(  
  
            stream_type = 3,  
  
            elementary_PID = 2071,  
  
            element_info_descriptor_loop = []  
  
        ),  
  
    ],
```



```
version_number = 1  
  
section_number = 0,  
  
last_section_number = 0,  
  
)
```

#Si se quisieran incluir aplicaciones interactivas en el segundo servicio se deberá modificar la *PMT* del servicio para indicar qué el servicio tiene aplicaciones.

#La *AIT* no será modificada ya que no se incluyen nuevas aplicaciones y sera identica a la del apartado 5.3.2.3.

#Si se quiere incluir información de eventos para el servicio se deben crear los objetos correspondientes a las secciones *EIT* donde se defina la información de eventos del segundo servicio.

#La *TDT* tampoco necesita modificaciones y sera identica a la del apartado 5.3.2.3.

#Por último se deben incluir las instrucciones que crearán el *Transport stream* de la tabla *PMT* "*firstpmt2.ts*" del nuevo servicio a partir de su objeto *pmt2*.

```
out = open("./pmt2.sec", "wb")  
  
out.write(pmt2.pack())  
  
out.close  
  
out = open("./pmt2.sec", "wb") # Python flush bug  
  
out.close  
  
os.system('/usr/local/bin/sec2ts ' + str(avalpa1_pmt_pid2) + ' <  
./pmt2.sec > ./firstpmt2.ts')
```

Realizadas todas estas modificaciones en el *script* se procede a su ejecución para obtener los *Transport streams* de las tablas *PSI*. Se obtendrán los mismos *Transport streams* que en el caso de un servicio mas el *Transport stream* correspondiente a la *PMT* del segundo servicio: *firtpmt2.ts*.

### 5.5.3.2 Puesta en marcha de Opencaster

La primera vez que se vaya a proceder a la emisión se deberán crear las todas las tuberías necesarias.

```
mkfifo      vlan1.ts      vlan2.ts      vlanfilter1.ts      vlanfilter2.ts  
vlan2remapped.ts      playout.ts      playouttdt.ts      playoutstamped.ts
```

- **Primer paso: Recepción de los contenidos**

En primer lugar se debe hacer notar que cada servicio será mandado por el VLC a una dirección *multicast* diferente.

**tsudpreceive 224.0.0.1 1234 > vlan1.ts &**

**tsudpreceive 224.0.0.2 1234 > vlan2.ts &**

- **Segundo paso: Filtrado de audio y vídeo**

Se debe eliminar la *PAT* y la *PMT* que introduce el VLC y tener en un *Transport stream* el audio y vídeo del servicio.

**tsfilter vlan1.ts +2064 +2068 > vlanfilter1.ts &**

**tsfilter vlan2.ts +2064 +2068 > vlanfilter2.ts &**

Se filtran las *PID*'s 2064 y 2068 porque son las que se han configurado en el VLC como *PID* de vídeo y audio. El VLC codifica los dos *Transport stream* con las mismas *PID*'s en cambio para el segundo servicio las *PID*'s definidas en la *PMT* son otras. Será necesario cambiar las *PID*'s por las especificadas en el *script* en *Python*.

- **Tercer paso: Sustitución de *PID*'s**

Para sustituir el valor del *PID* de un *Transport stream* se hace uso del comando *tspidmapper*.

**tspidmapper vlanfilter2.ts 2064 to 2070 and 2068 to 2071 > vlan2remapped.ts &**

De esta forma se tendrá en *vlan2remapped.ts* el audio y vídeo del segundo servicio con los *PID*'s que tienen definidos en *firstpmt2.ts*.

- **Cuarto paso: Multiplexación del *Transport stream***

Suponiendo que los contenidos han sido codificados de la misma forma con el *Reference* de *Mainconcept* y ya se ha averiguado su tasa de bit, se procede a la *multiplexación* de los dos servicios con los *Transport stream* de sus tablas y el *Transport stream* de la aplicación del primer servicio.

**tscbrmuxer b:4795504 vlanfilter1.ts b:4795504 vlan2remapped.ts b:1000000 mhp.ts b:10000 firstait.ts b:10000 firstpat.ts b:10000 firstpmt.ts b:10000 firstpmt2.ts b:10000 firstsdt.ts b:10000 firstnit.ts b:10000 firsteit.ts b:10000 firsttdt.ts b: 9234992 null.ts > *playout.ts* &**

Los siguientes pasos son los mismos que para un solo servicio.

- **Quinto paso: Corrección de la hora**

Como se pretende que el *Transport stream* lleve la hora actual, se actúa sobre el *Transport stream* con el comando *tstdt* para ese propósito.

**tstdt playout.ts > playouttdt.ts &**

- **Sexto paso: Corrección de las marcas de referencia de reloj**

Se corrigen las marcas de referencia de reloj con el comando *tspcrstamp*.

**tspcrstamp playouttdt.ts 19906000 > playoutstamped.ts &**

- **Septimo y ultimo paso: Emisión**

Finalmente se pone en emisión el *Transport stream* multiplexado *playoutstamped.ts*.

**DtPlay playoutstamped.ts -t 110 -mt OFDM -mC QAM64 -mG 1/4 -mc 2/3 -mf 578**

El *script* completo queda:

**tsudpreceive 224.0.0.1 1234 > vlan1.ts &**

**tsudpreceive 224.0.0.2 1234 > vlan2.ts &**

**tsfilter vlan.ts +2064 +2068 > vlanfilter.ts &**

**tsfilter vlan1.ts +2064 +2068 > vlanfilter1.ts &**

**tsfilter vlan2.ts +2064 +2068 > vlanfilter2.ts &**

**tspidmapper vlanfilter2.ts 2064 to 2070 and 2068 to 2071 > vlan2remapped.ts &**

**tscbrmuxer b:4795504 vlanfilter1.ts b:4795504 vlan2remapped.ts b:1000000 mhp.ts b:10000 firstait.ts b:10000 firstpat.ts b:10000 firstpmt.ts b:10000 firstpmt2.ts b:10000 firstsdt.ts b:10000 firstnit.ts b:10000 firsteit.ts b:10000 firsttdt.ts b: 9234992 null.ts > playout.ts &**

**tstdt playout.ts > playouttdt.ts &**

**tsstamp playouttdt.ts 19906000 > playoutstamped.ts &**

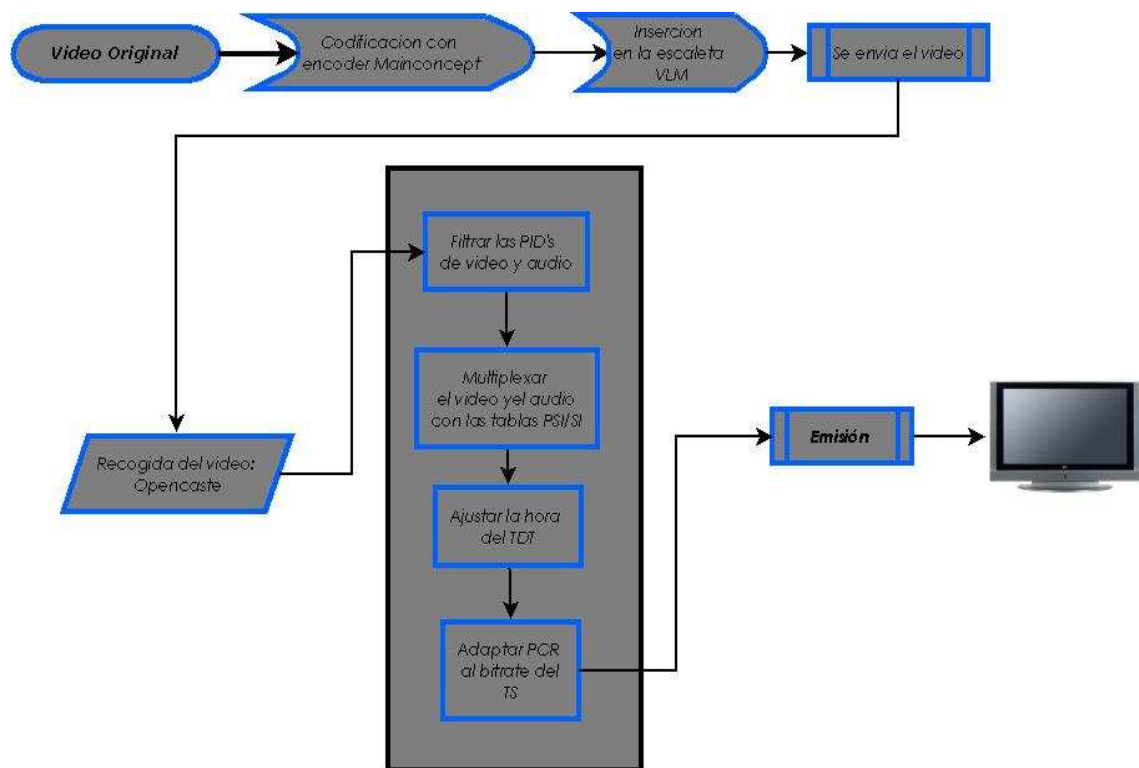
**DtPlay playoutstamped.ts -t 110 -mt OFDM -mC QAM64 -mG 1/4 -mc 2/3 -mf 578**



## 6 Codificación y adaptaciones del vídeo

### 6.1 Introducción

Hasta ahora se ha hablado de cómo realizar la automatización del canal de televisión digital y de cómo usar las herramientas de *Opencaster*. Pero el texto no se ha centrado en las transformaciones que tiene el vídeo. Cada herramienta que se ha utilizado tiene una consecuencia en el vídeo, que le lleva a tener una serie de transformaciones, que tienen como objetivo que el vídeo pueda ser interpretado, y por lo tanto visto, en los receptores finales del sistema. El proceso que sigue el vídeo viene ilustrado en la *figura 33*. El esquema parte desde lo que se considera el vídeo original, que viene a ser el vídeo captado con las cámaras, en el formato nativo de estas, hasta que este es modulado y emitido.



**Figura 33**

Como se ha comentado en el apartado 5.6, el *Opencaster* necesita que la tasa de bit del vídeo entrante sea constante. Para eliminar una posible variabilidad en la tasa de bit, la cual nos impediría utilizar el *Opencaster*, es indispensable realizar una codificación previa del vídeo a emitir. Esto nos permite tener una tasa de bit constante y conocida. Vamos a ver como realizar esta codificación.

## 6.2 Codificación

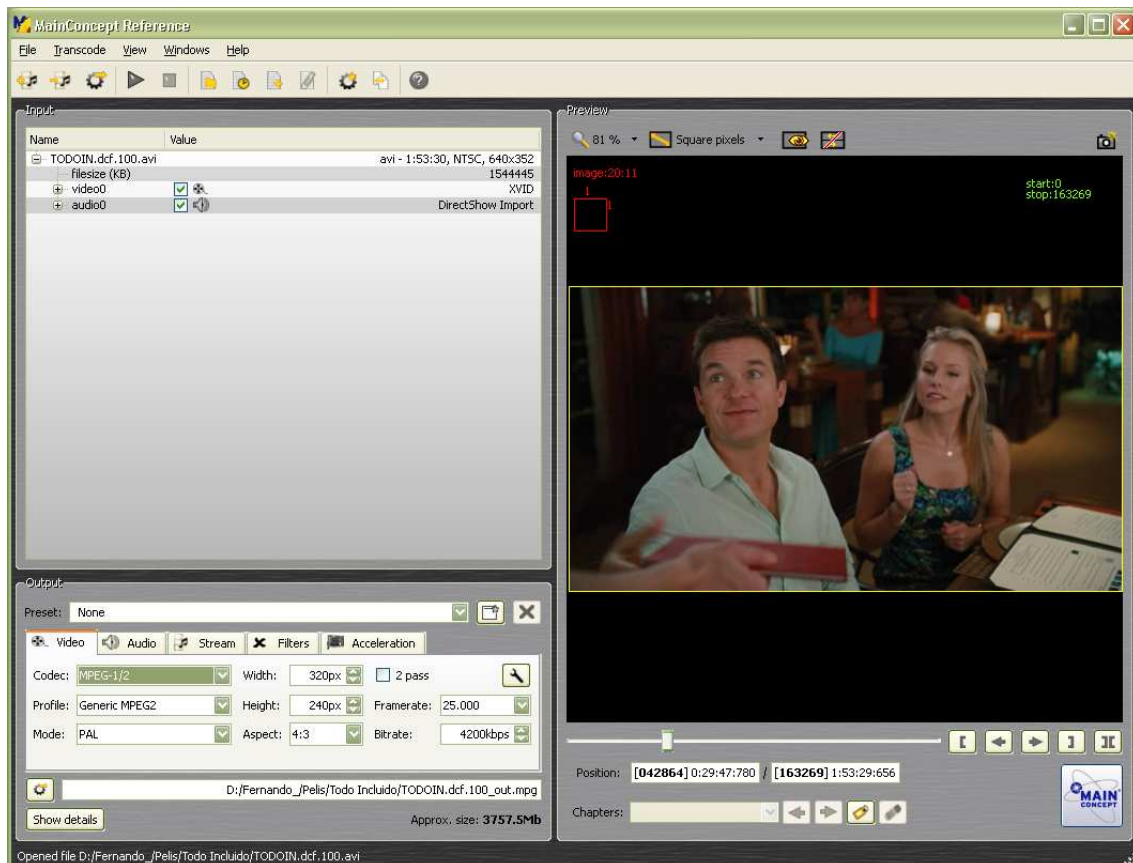
La codificación se realizará con el software *Mainconcept Reference*. Este software tiene una licencia privada y por lo tanto es de pago, pero es una casa altamente reconocida en el mercado, además, su producto tiene gran capacidad para variar diversos parámetros. De hecho, este software es utilizado por productos de edición de vídeo como *Adobe Premiere CS4*, y podría ser adquirido como parte de un paquete más grande.

Tener que utilizar una licencia privada en este punto, es una opción que se plantea por la sencillez y potencia que ofrece este software. No es la única opción, y el mercado si dispone de software de libre distribución que puede hacer la codificación. El *encoder ffmpeg* es de libre distribución y sirve para el sistema propuesto. Si se opta por *Mainconcept reference* es por sus buenas características, y por no ser un precio demasiado elevado. Además, cuando se edita el vídeo que se quiere emitir, la exportación se puede realizar con las características que se describen a continuación, solventando el problema de la codificación.

Otro problema que surge por el hecho de tener que utilizar una codificación previa, es que no se podrá hacer una emisión en directo, con el sistema tal cual descrito hasta ahora. Esto se podría arreglar añadiendo un modulo que hiciera la codificación en vivo. La casa ABE dispone de módulos que hacen este trabajo, dando una salida con tasa de bit constante y conocida, con lo cual se podría seguir utilizando en *Opencaster*, si en un futuro, se quisiera hacer emisiones en directo.

El paso de codificación del vídeo pretende solventar dos problemas a la vez: adaptar la señal a la codificación que utiliza el estándar *DVB* y conseguir que todos los archivos de salida tengan exactamente la misma *tasa de bit*. Esto segundo es de gran importancia, ya que las herramientas utilizadas del *Opencaster* necesitan que todos los vídeos tengan igual *tasa de bit*, como se explicará más adelante.

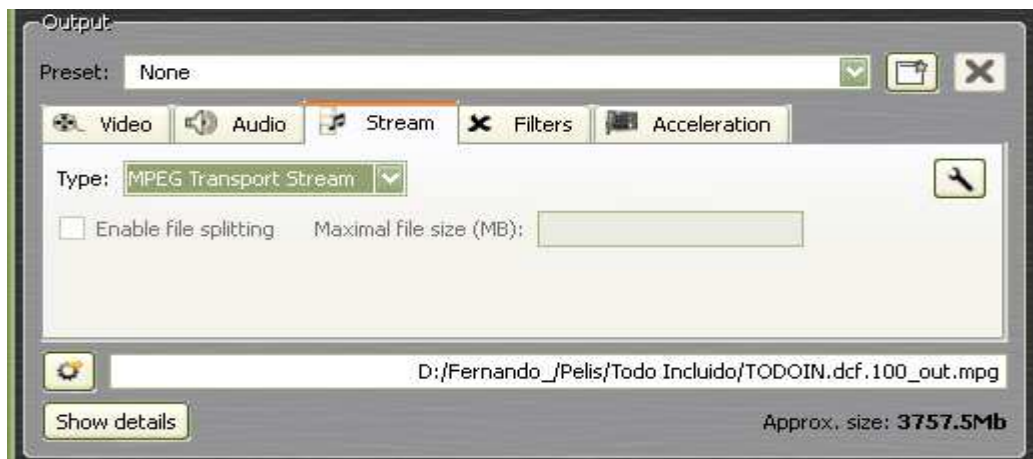
El aspecto del programa es el siguiente:



**Figura 34**

El software dispone de muchas opciones para codificar. Aquí se explicarán paso a paso las opciones que se deben activar para obtener la codificación que se desea para el caso de estudio.

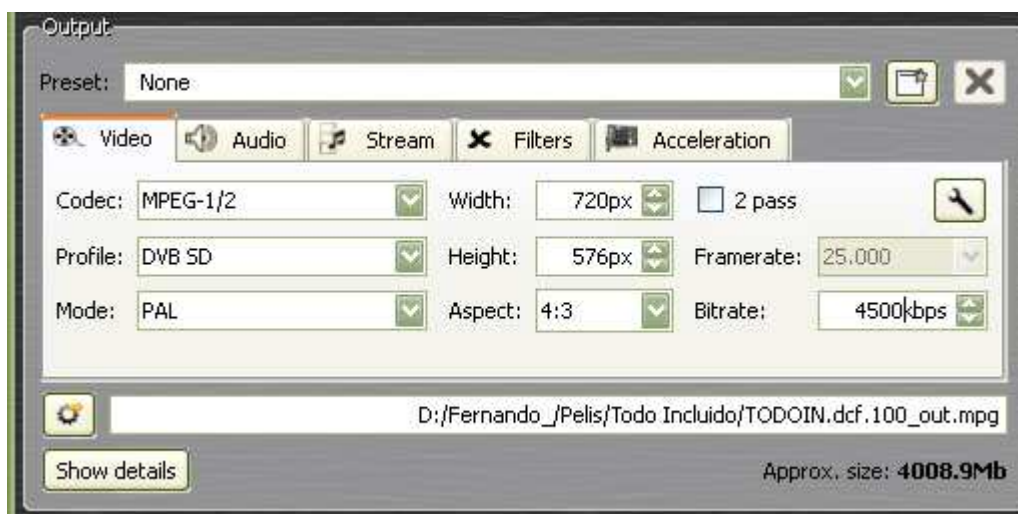
Se debe obtener de salida una trama *Transport stream*, ya que es un vídeo para emisión de televisión digital. Así que lo primero es elegir que la salida debe ser del tipo *MPEG Transport stream*. En la pestaña de *Stream*, como se ve en la figura 35:



**Figura 35**

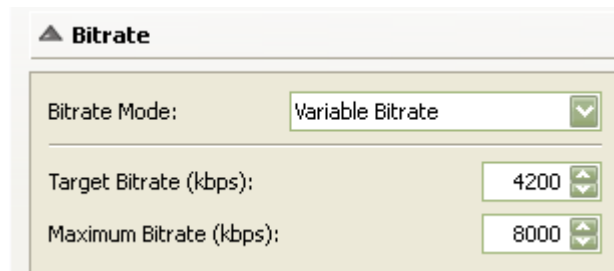
Si se presiona el botón de opciones se pueden cambiar diversos parámetros del *Transport stream* de salida, pero para el caso de estudio la configuración predeterminada, es adecuada.

Lo siguiente será ver las opciones de la codificación de vídeo. Se presiona en la pestaña de vídeo. El *codec* que se utiliza es *MPEG-1/2* con el tipo de *DVB Standard*. Va a ser una transmisión *PAL*. Una vez que se ha seleccionado el tipo de codificación, las opciones de ancho, alto y el aspecto, cambian al estándar requerido, es decir, 720x576 y 4:3 de aspecto. Solo queda por seleccionar que la tasa de bit sea constante. Esto se realiza presionado el botón de opciones, en la pestaña de *Bitrate*. Aquí se ha usado un tasa de bit constante de 4500 Kbps., aunque el valor podría ser cualquiera que este por encima de 3825 kbps, que es el valor mínimo que marca el *Mainconcept* para este tipo de codificación. La configuración final debe ser la que se aprecia en la *figura* siguiente:



**Figura 36**





**Figura 37**

Lo ultimo que queda es cambiar las opciones de audio. Se presiona la pestaña de *Audio*. El *codec* de audio que se utiliza es *MPEG audio* y canal estereo. El *bitrate* se cambia a 128 Kbps. El aspecto es el siguiente:



**Figura 38**

Con esto, quedan configurados todas las opciones de codificación de transcodificación y se puede obtener el vídeo de salida, dando al botón de *Start transcode*. Este vídeo que obtenemos es el que posteriormente se inserta en la escaleta *VLM*

## 6.3 Adaptaciones del video

### 6.3.1 Inserción en la escaleta y emisión

El proceso de inserción en la escaleta ya ha sido descrito anteriormente, el capítulo 4 está dedicado enteramente a esta tarea, con lo cual no se comentará en detalle este proceso, salvo que la emisión que se hará con el VLC será sin transcodificación, ya que estos ya están codificados adecuadamente como se acaba de describir.

Lo importante aquí es que, durante este proceso, los *PID's* de vídeo y de audio se modifican. En el proceso de codificación con *Mainconcept* no se pueden elegir estos *PID's*, si no que *Mainconcept* coloca por defecto unos. En concreto para el video el 460 y para el audio el 462, así que es con el VLC donde se elige estos valores. Los valores en los que se recibe con el *Opencaster* son 2064 para el vídeo y 2068 para el audio.

Estos 2 valores, son los valores que vienen por defecto en el *Opencaster*, aunque evidentemente se pueden variar en el *script* de *Python*. Por sencillez se ha preferido cambiar los *PID's* con el VLC en vez de en el *script*.

Para elegir estos valores de salida del *Transport stream* hay que ir a preferencias del VLC y en la sección de "salida de emisión/Muxores/MPEG-TS" cambiar por los valores deseados. Esto queda ilustrado en la *Figura 39*.

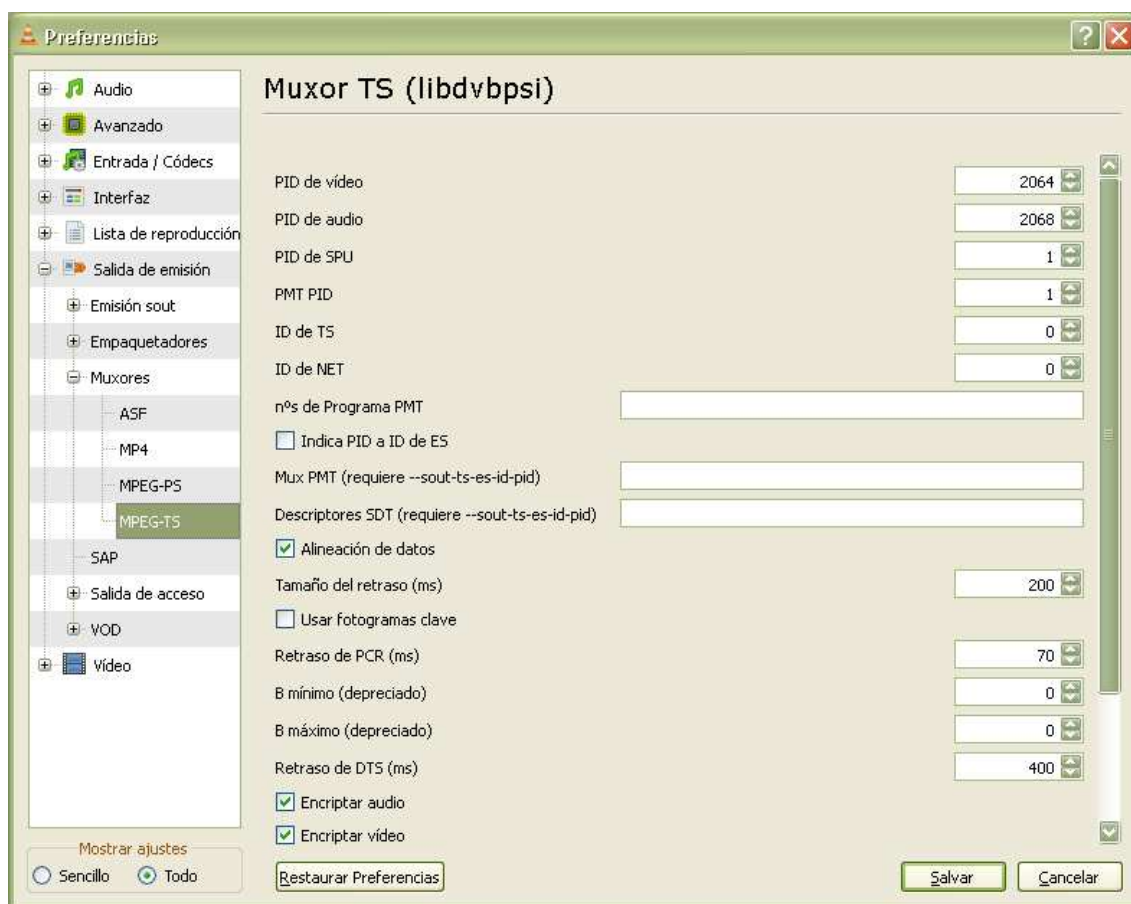


Figura 39

### 6.3.2 Recogida del vídeo: Opencaster

Una vez emitido el vídeo, este es recogido por el *Opencaster*. Aquí solo se va a describir las adaptaciones que sufre el vídeo. Los métodos de cómo realizarlo y cómo deben ser utilizados los comandos necesarios, ya han sido comentados, así que no se entrará más en detalle. Lo primero que se realiza es:

- **Filtrar *PID*'s de vídeo y audio**

El primer paso es quedarse solo con los *PID*'s de vídeo y audio, descartando el resto del *Transport stream*. Esto se realiza con el comando del *Opencaster* *tsfilter*. La salida será un *Transport stream* que solo contiene el vídeo y el audio.

- **Multiplexar el *Transport stream* con tablas *PSI-SI***

En este proceso, se le añade al *Transport stream* de vídeo, todas las tablas *PSI-SI* que previamente han sido creadas y empaquetadas en diferentes *Transport stream*. Además, también se le multiplexa una aplicación interactiva *MHP*. Esto se realiza con el comando de *Opencaster* *tsmuxer*. Este comando generará a la salida un *Transport stream* que contenga el vídeo, audio, tablas *PSI-SI* y la aplicación interactiva.

- **Ajustar la hora de la tabla TDT**

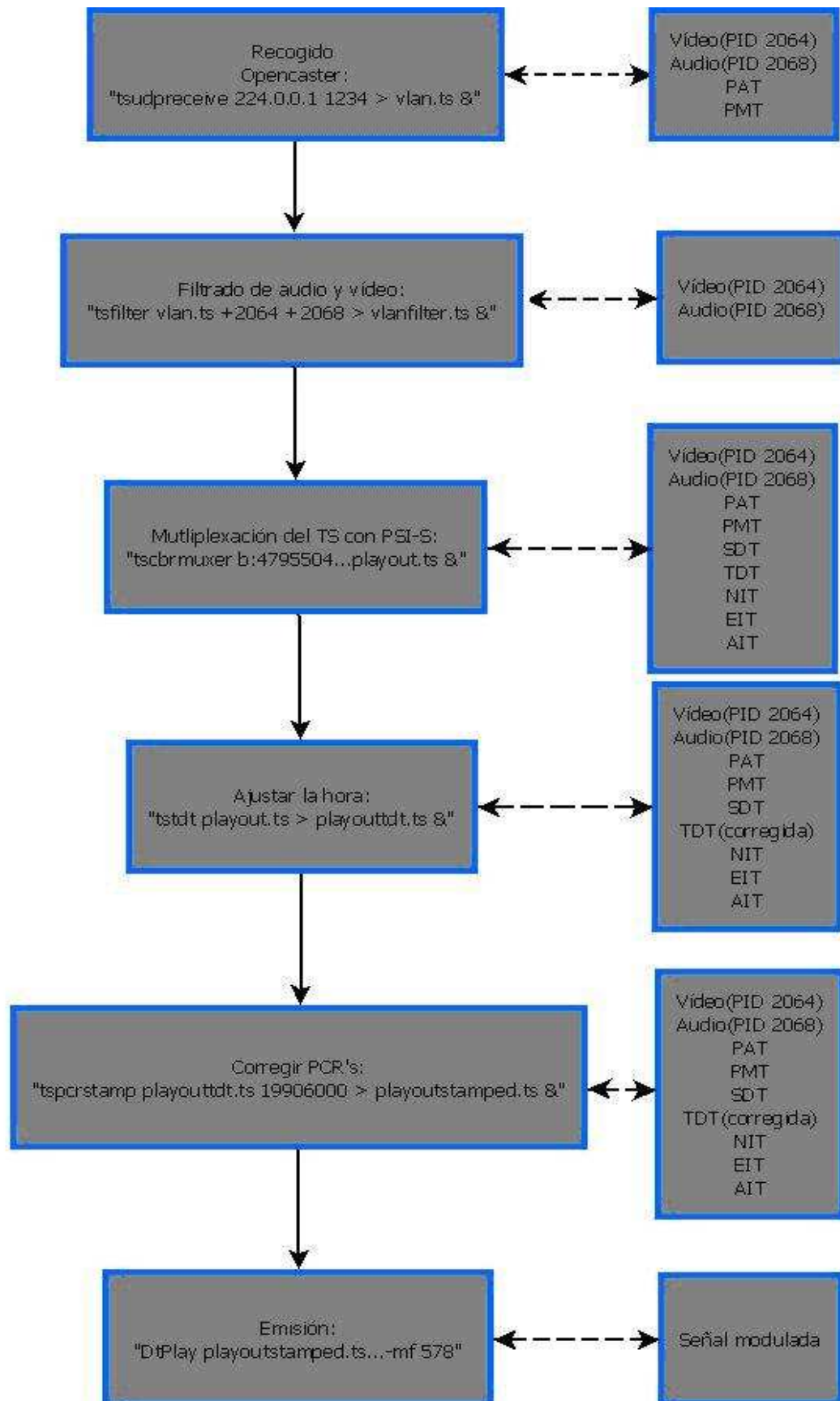
Ahora se pasa a modificar la tabla *TDT* para actualizar la hora. Esto se realiza con el comando *tstdt*, y con el si consigue actualizar la hora con la que aparece en el ordenador.

- **Adaptar el PCR al bitrate del Transport stream**

Con el comando *tspcrstamp* se arreglan las marcas de de referencia de reloj. Si no se hace esta modificación la visualización no será correcta después en pantalla, con pequeños fallos de sincronía entre audio y video.

Este es la ultima adaptación que necesita el archivo. El vídeo ya esta listo para que la tarjeta *Dektec* module la señal y llegue a los receptores de *TDT*.

En la *Figura 40* se puede observar todo el proceso:



**Figura 40**

**Figura 38**

## **7 Conclusiones y recomendación final**

A lo largo del presente documento se han presentado diferentes soluciones para la infraestructura necesaria a la hora de implantar un canal de televisión propio en el Valle de Aranguren, cumpliendo con uno de los objetivos fundamentales fijados en las especificaciones del proyecto de colaboración firmado entre el Ayuntamiento del Valle de Aranguren, la UPNA y los CES.

El espíritu de este documento ha sido, en todo momento, didáctico; con el objetivo de dar las herramientas necesarias para que la decisión tomada por el Ayuntamiento pueda darse de manera razonada en base a un conocimiento real de cuales son las funcionalidades y limitaciones del sistema que está implantando.

La conclusión más triste del proyecto, sin duda, es que el ambicioso proyecto que subyacía entre los objetivos de la UPNA y de los CES, como era el de la puesta en marcha de una prueba piloto en televisión digital interactiva de última generación, no va a poder ser llevado a la práctica, al menos a corto plazo. Este proyecto piloto tenía como objetivo crear una red de televisión capaz de responder a los avances que se producirán en televisión digital interactiva a medio plazo, como son la emisión de contenidos en alta definición, la flexibilidad en los sistemas de acceso condicional emitidos y la implementación de aplicaciones *MHP* de última generación. Sin embargo, la ausencia de receptores de televisión digital en el mercado (ni tan siquiera a nivel experimental) han hecho imposible la consecución de dicho objetivo. A la UPNA y a los CES les queda el consuelo de haberlo intentado por todos los medios y de haber topado con una limitación tecnológica que no podía ser superada. Además, dado que esta meta no estaba entre los objetivos del proyecto, tampoco se puede hablar de fracaso en ese sentido, pero la espinita queda ahí clavada.

Entrando a resumir las que pueden ser las recomendaciones principales que se derivan de este documento, se comienza por resumir las principales conclusiones del estudio realizado:

- La interactividad en televisión es una plataforma de acercamiento al ciudadano con un potencial único y que merece la pena ser considerada.
- Gracias a la solución propuesta en el apartado 2.3.3 (Servidor MHP recibiendo vídeo por streaming), el coste de insertar interactividad es prácticamente el mismo que el de no insertarla.
- La implantación de la solución propuesta en el apartado 2.3.3 es independiente de la decisión del Ayuntamiento de Aranguren en relación al desarrollo a corto plazo de las aplicaciones MHP, estando la infraestructura preparada para incluirlas en el momento que se considere oportuno. Además, esta solución es escalable de un modo natural en caso de que en un futuro el Ayuntamiento de Aranguren

pretenda ampliar sus instalaciones mejorando todo el proceso de producción audiovisual.

Por otro lado, y aunque no se ha entrado en este tema a lo largo del documento, es importante hacer hincapié en la importancia de una instalaciones debidamente preparadas para alojar la cabecera de televisión digital. En este sentido, es altamente recomendable la mejora de las instalaciones actuales mediante:

- La adquisición de un sistema de aire acondicionado que asegure una temperatura mínima en las instalaciones de cabecera, ya que este es un parámetro crítico para el correcto mantenimiento de los equipos informáticos que serán alojados en las mismas.
- La adquisición de un SAI (Sistema de Alimentación Ininterrumpida) que alimente a los equipos informáticos o incluso a toda la cabecera. Un SAI es un dispositivo que gracias a sus baterías, puede proporcionar energía eléctrica tras un apagón a todos los dispositivos que tenga conectados. Otra de las funciones de los SAI es la de mejorar la calidad de la energía eléctrica que llega a los aparatos, filtrando subidas y bajadas de tensión y eliminando armónicos de la red en el caso de usar corriente alterna. Como regla general, los SAI dan energía eléctrica a equipos llamados *cargas críticas*, como pueden ser aparatos médicos, industriales o informáticos que, como se ha dicho antes, requieren tener siempre alimentación y que ésta sea de calidad, debido a la necesidad de estar en todo momento operativos y sin fallos. Probablemente, considerar la televisión un servicio fundamental es ir un poco lejos en las atribuciones de la misma, pero en cualquier caso, esta es una mejora a contemplar en un futuro.

Si bien es cierto, como se ha comentado, que el proyecto que subyacía entre los objetivos de la UPNA y de los CES, que era el de la puesta en marcha de una prueba piloto en televisión digital interactiva de última generación, no va a poder ser llevado a la práctica, al menos a corto plazo, el segundo objetivo de este proyecto, como es el de montar un canal de televisión digital interactiva mediante software de libre distribución, si se ha cumplido, y se puede concluir que es posible disponer de un canal de televisión interactiva con una inversión mínima a base de utilizar software de libre.

Con la salvedad de que son programas no demasiado intuitivos, se requiere de cierto tiempo para aprender su correcto funcionamiento y no disponen de todas las herramientas de las que disponen los programas de pago, siendo una de las limitaciones mas acusadas la complejidad en la creación de las escaletas de continuidad, la solución aportada se podría adaptar bien a un canal pequeño, sin demasiada producción propia, como viene a ser la televisión que se planteo montar en un principio en el valle de Aranguren, en la propuesta que se hizo por parte de estos, a la UPNA. Así que la conclusión que se saca es positiva, y esta solución podría ser implementada



en un futuro, en pequeñas localidades que quisieran disponer de un canal de televisión digital interactivo, a un coste mínimo.

La única salvedad en cuanto a software libre, es haber tenido que usar un software de pago, *Reference* de *Mainconcept*, para la codificación previa de los videos. A pesar de ser de pago existe una versión de prueba que permite que todo el procesado de los contenidos para su emisión tenga coste cero. Si bien esta versión de prueba implica que los videos a emitir lleven el logo de *Mainconcept* incrustado.

Si a pesar de esto no se quisiera comprar este paquete, este escollo se podría solventar con el software *ffmpeg* y con su interfaz gráfica de usuario *WinFF*, que si es de libre distribución y además de ser un encoder bastante potente.

En cualquier caso el coste es mínimo en comparación con cualquier otra opción. Además, el *encoder* de *Mainconcept* viene integrado en programas de edición de vídeo como, *Adobe CS4*. Estos programas de edición de vídeo serán imprescindibles de adquirir si se quiere montar un canal de televisión digital, así que no supone un gran problema.

Los mayores problemas que se pueden destacar son que ciertos aspectos en el uso del canal pueden resultar incómodos de cara al usuario final. Sobre todo el uso y configuración de *Opencaster*. Además, la interfaz web del *VLM* puede llegar a no ser del todo intuitiva a la hora de crear una escaleta diaria y menos incluso una escaleta semanal.

Lo anteriormente comentado es sugerido como posible mejora para la implementación del canal de televisión con la ayuda del *VLC* y *Opencaster*, y puede resultar como propuesta de posibles proyectos futuros.

En primer lugar sería deseable poder contar con un interfaz gráfico de usuario para el manejo de las herramientas de *Opencaster*. Con dicho interfaz se pretende un manejo más intuitivo y sencillo por parte del usuario, sin el uso de la línea de comandos de Linux. También se buscaría poder modificar los distintos parámetros del *script* en *Python* requeridos a la hora de emitir el canal, con un entorno visual, como por ejemplo:

- La configuración de los *PID's* de los distintos servicios.
- La configuración de las aplicaciones *MHP*.
- Una más fácil inclusión de la información de la *EPG*.

Otro aspecto a mejorar sería el propio interfaz web del *VLM* propio del *VLC*. En ese sentido sería conveniente que el interfaz fuera encaminado a una escaleta de contenidos de forma diaria y semanal, añadiendo la funcionalidad de la concatenación de contenidos de forma automática.

En cualquier caso aunque las mejoras que se pueden hacer son evidentes, para dar un servicio, en que no se pretende hacer una emisión de contenidos de 24 horas, ó en que se pretende tener videos en bucle durante cierto tiempo, el sistema se puede considerar bastante adecuado, ya que



cumple los requerimientos básicos y es un sistema estable. Todo ello con coste muy limitado.

Finalmente, un breve comentario sobre una cuestión administrativa que tampoco se ha mencionado en el documento. La Ley 10/1988, de 3 de mayo, de Televisión Privada, obliga a las cadenas de televisión a almacenar una copia legal de la producción emitida durante todos los días y en todas las horas por un periodo de seis meses. En principio, existen equipos destinados a realizar esta copia legal, los cuales están preparados para sobrescribir el contenido cuyo almacenamiento ya no es necesario (una vez que han pasado seis meses desde su emisión) e ir sustituyéndolo paulatinamente con el contenido en emisión. Estos sistemas pueden ser muy costosos o relativamente asequibles. La mayoría de las empresas especializadas plataformas de continuidad (*VSN, Pebble Beach, Vector3...* etc.) disponen de un equipo destinado a este fin y que se integra de manera automática en su sistema, por lo que, en caso de que en un futuro se considere conveniente, su integración en la cabecera de televisión del Valle de Aranguren sería relativamente sencilla. Sin embargo, dado el escaso contenido propio que en un principio será producido y emitido por el Valle de Aranguren, parece del todo sobredimensionada la adquisición de uno de estos equipos. En definitiva, bastará con tener en cuenta que todo el contenido que se emita, debe ser correctamente catalogado de tal forma que en caso de producirse alguna reclamación sea posible la recuperación de dicho contenido.

## Referencias

- [Abe1] CD Emp. ABE: Hoja de características de la familia de productos de la casa ABE ("Abe1 Hoja características DME-EMX-DVM-MUX1000.pdf").
- [Abe2] CD Emp. ABE: Hoja de características de la familia de productos de la casa ABE ("Abe2 Especificaciones DVM1000 Agile Up Converter.pdf").
- [Abe3] CD Emp. ABE: Hoja de características de la familia de productos de la casa ABE ("Abe3 Especificaciones ABE MPEG2 Multiplexer Board.pdf").
- [Abe4] CD Emp. ABE: Hoja de características de la familia de productos de la casa ABE ("Abe4 Especificaciones DME1000.pdf").
- [Abe5] CD Emp. ABE: Hoja de características de la familia de productos de la casa ABE ("Abe5 Especificaciones DVM1000TH.pdf").
- [Abe6] CD Emp. ABE: Hoja de características de la familia de productos de la casa ABE ("Abe6 Especificaciones EMX1000.pdf").
- [Abe7] CD Emp. ABE: Hoja de características de la familia de productos de la casa ABE ("Abe7 Especificaciones MUX1000.pdf").
- [AbeP1] CD Emp. ABE: Oferta para la familia de productos de la casa ABE ("AbeP1 Oferta oe511U-09 multiplexor.pdf").
- [AbeP2] CD Emp. ABE: Hoja de características de la familia de productos de la casa ABE ("AbeP2 Oferta oe512U-09 encoders.pdf").
- [ActP1] CD Emp. Activa Multimedia: Resumen y oferta de la plataforma para la automatización de la continuidad del canal ("ActP1 CANAL CIUDADANO.ppt").
- [Adb1] CD Emp. ADB: Especificaciones del receptor modelo DTR8740 ("Ab1 especificaciones DTR 8740.pdf").
- [Adb2] CD Emp. ADB: Especificaciones del receptor modelo 3810T ("Abd2 especificaciones modelo 3800\_10T").
- [Adb3] CD Emp. ADB: Especificaciones del receptor modelo 5100T ("Adb3 especificaciones modelo 5100TX.pdf").
- [Adb4] CD Emp. ADB: Especificaciones del receptor modelo 5810T ("Adb4 especificaciones modelo 5810TX.pdf").
- [Adh1] [www.adht.com](http://www.adht.com)
- [Ait1] Cd Memoria Documentacion/ETSI TS 102 812 V1.1.1.pdf 10.4.6

- [Ait2] Cd Memoria Documentacion/ETSI TS 102 812 V1.1.1.pdf 10.2.1
- [Ait3] Cd Memoria Documentacion/ ETSI TS 101 162 V1.2.1.pdf 5.7 Tabla 15
- [Alt1]] <http://www.alticast.com/main.html>
- [App1] CD Emp. AppearTv : Especificaciones del multiplexor de la casa AppearTv ("App1 especificaciones MC3000.pdf").
- [AppP1] CD Emp. AppearTv : oferta del multiplexor de la casa AppearTv ("AppP1 especificaciones MC3000.pdf").
- [Ara1] Documentación entregada al Ayuntamiento del Valle de Aranguren como resultado del Proyecto OTRI 2006 12 128, "Red de telecomunicación valle de Aranguren".
- [Ce1] [www.cein.es](http://www.cein.es)
- [Ces1] [www.cesnavarra.net](http://www.cesnavarra.net)
- [Ces2] Documento: "Propuesta de servicios para el Valle de Aranguren", elaborado por los Centros de Excelencia Software de Navarra y la Universidad Pública de Navarra, a cargo del proyecto *OTRI 2009024051 "Propuesta de soluciones para la inserción de un canal de televisión propio del ayuntamiento en la red de cable del valle de Aranguren y de posibles aplicaciones de televisión digital interactiva a desarrollar"*.
- [Cin1] CD Emp. Cinegy: Especificaciones de los productos de la familia Cinegy ("Cin1 Cinegy\_workflow\_brochure.pdf").
- [Cin2] CD Emp. Cinegy: Especificaciones de los productos de la familia Cinegy ("Cin2 Cinegy\_eXpress\_brochure.pdf").
- [Cin3] CD Emp. Cinegy: Especificaciones de los productos de la familia Cinegy ("Cin3 cinegy\_air.pdf").
- [Cin4] CD Emp. Cinegy: Especificaciones de los productos de la familia Cinegy ("Cin4 cinegy\_ingest.pdf").
- [Cin5] CD Emp. Cinegy: Especificaciones de los productos de la familia Cinegy ("Cin5 Cinegy\_media\_desktop.pdf").
- [Cin6] CD Emp. Cinegy: Especificaciones de los productos de la familia Cinegy ("Cin6 cinegy\_media\_archive.pdf").
- [Cin7] CD Emp. Cinegy: Especificaciones de los productos de la familia Cinegy ("Cin7 Especificaciones requerimientos del sistema (PC).pdf").

- [Cof1] Información en Wikipedia sobre COFDM  
<http://es.wikipedia.org/wiki/COFDM>
- [Dek1] CD Emp. Dektec: Especificaciones de los productos de la familia Dektec ("Dek1 Dektec Catalogo 2009.PDF").
- [Dek2] CD Emp. Dektec: Especificaciones de los productos de la familia Dektec ("Dek2 Dektec DTA-110T DVB COFDM Modulator.pdf").
- [Dek3] CD Emp. Dektec: Especificaciones de los productos de la familia Dektec ("Dek3 Dektec DTA-112 DVB COFDM Modulator with ASI input.pdf").
- [Dek4] CD Emp. Dektec: Especificaciones de los productos de la familia Dektec ("Dek4 Dektec DTA-2137 DVB Satellite input.pdf").
- [Dek5] CD Emp. Dektec: Especificaciones de los productos de la familia Dektec ("Dek5 Dektec DTC-700 MuxXpert.pdf").
- [Dek6] CD Emp. Dektec: Especificaciones de los productos de la familia Dektec ("Dek6Dektec DTU-225 ASI input.pdf").
- [Dek7] CD Emp. Dektec: Especificaciones de los productos de la familia Dektec ("Dek7 Dektec DTA-105 ASI Output.pdf").
- [Dek8] CD Emp. Dektec: Especificaciones de los productos de la familia Dektec ("Dek8 Dektec DTC-300 StreamXpress Leaflet.pdf").
- [DekP2] CD Emp. Dektec: Oferta de los productos de la familia Dektec ("DekP2 Dektec DTA-110T DVB COFDM Modulator.pdf").
- [DekP3] CD Emp. Dektec: Oferta de los productos de la familia Dektec ("DekP3 Dektec DTA-112 DVB COFDM Modulator with ASI input.pdf").
- [DekP4] CD Emp. Dektec: Oferta de los productos de la familia Dektec ("DekP4 Dektec DTA-2137 DVB Satellite input.pdf").
- [DekP5] CD Emp. Dektec: Oferta de los productos de la familia Dektec ("DekP5 Dektec DTC-700 MuxXpert.pdf").
- [DekP6] CD Emp. Dektec: Oferta de los productos de la familia Dektec ("DekP6 Dektec DTU-225 ASI input.pdf").
- [DekP7] CD Emp. Dektec: Oferta de los productos de la familia Dektec ("DekP7 Dektec DTA-105 ASI Output.pdf").
- [Dsm1] Información en Wikipedia sobre el carrusel de objetos DSM-CC  
<http://es.wikipedia.org/wiki/DSM-CC>
- [Dvb1] [www.dvb.org](http://www.dvb.org)

- [Dvb2] Especificación de la modulación DVB-T. "DVB BlueBook A122 Rev.1, Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)",  
<http://www.dvb.org/technology/standards/a122r1.tm3980r7.DVB-T2.pdf>
- [Eit1] Cd Memoria Documentacion/ETSI-EN-300-468.pdf 5.2.4
- [Epg1] Información en Wikipedia sobre EPG  
[http://es.wikipedia.org/wiki/Gu%C3%ADa\\_Electr%C3%B3nica\\_de\\_Programas](http://es.wikipedia.org/wiki/Gu%C3%ADa_Electr%C3%B3nica_de_Programas)
- [Els1] Información en Wikipedia sobre los *Elementary stream*  
[http://en.wikipedia.org/wiki/Elementary\\_stream](http://en.wikipedia.org/wiki/Elementary_stream)
- [Eur1] CD Emp. Eurotek: Especificaciones de los productos de la familia Eurotek ("Eur1 Plataforma All4digit.pdf").
- [Eur2] CD Emp. Eurotek: Especificaciones de los productos de la familia Eurotek ("Eur2 multiplexador EK AMX\_4.pdf").
- [EurP1] CD Emp. Eurotek: Oferta para los productos de la familia Eurotek ("EurP1 Oferta Eurotek\_09582 modulador RF.pdf").
- [EurP2] CD Emp. Eurotek: Oferta para los productos de la familia Eurotek ("EurP2 Primera Oferta Eurotek\_09575 Modulador IF.pdf").
- [Gra1] CD Emp. *Grassvalley*: Especificaciones de los productos de la familia grassvalley ("Gra1 especificaciones Grass Valey Multiplexor Amber II.pdf").
- [Gra2] CD Emp. *Grassvalley*: Especificaciones de los productos de la familia grassvalley ("Gra2 Grass Valey Coral MHP Broadcasting server.pdf").
- [Gra3] CD Emp. *Grassvalley*: Especificaciones de los productos de la familia grassvalley ("Gra3 Grass Valey NetProcesor9030.pdf").
- [GraP1] CD Emp. *Grassvalley*: Especificaciones de los productos de la familia grassvalley ("GraP1 Oferta CORAL\_AmberII\_v2\_091008\_customer.xls").
- [GraP2] CD Emp. *Grassvalley*: Especificaciones de los productos de la familia grassvalley ("GraP2 Primera Oferta agosto N090811MDA1\_U\_NAVARRA\_CORAL\_9030\_v1\_090811\_client.xls").
- [Hps1] información sobre la cabecera de un PES  
<http://dvd.sourceforge.net/dvdinfo/pes-hdr.html>.

- [Hum1] CD Emp. Ican-t: Especificaciones sobre receptores de la casa Humax("Hum1 Especificaciones tuvibox.pdf").
- [Hum2] CD Emp. Ican-t: Especificaciones sobre receptores de la casa Humax("Hum2 Especificaciones DTT3600.pdf").
- [Ibc1] [www.ibc.org](http://www.ibc.org)
- [Ica1] CD Emp. Icareus: Especificaciones de los productos de la familia Icareus ("Ica1 Appendix\_1\_Icareus\_iTV\_Suite\_Author\_Datasheet.pdf").
- [Ica2] CD Emp. Icareus: Especificaciones de los productos de la familia Icareus ("Ica2 Appendix\_2\_iTV\_Suite\_Player\_Datasheet.pdf").
- [Ica3] CD Emp. Icareus: Especificaciones de los productos de la familia Icareus ("Ica3 Appendix\_3\_iTV\_Suite\_Viewer\_Datasheet.pdf").
- [Ica4] CD Emp. Icareus: Especificaciones de los productos de la familia Icareus ("Ica4 Appendix\_4\_iTV\_Suite\_Integrator\_Datasheet.pdf").
- [Ica5] CD Emp. Icareus: Especificaciones de los productos de la familia Icareus ("Ica5 Appendix 5 iTV Suite Component Pack Datasheet.pdf").
- [Ica6] CD Emp. Icareus: Especificaciones de los productos de la familia Icareus ("Ica6 Appendix\_6\_Icareus\_Playout\_PC100\_Compact Datasheet.pdf").
- [Ica8] CD Emp. Icareus: Especificaciones de los productos de la familia Icareus ("Ica8 Appendix\_8\_Icareus\_Playout EP100 EPG Server Datasheet.pdf").
- [Ica9] CD Emp. Icareus: Especificaciones de los productos de la familia Icareus ("Ica9 Appendix\_9\_Icareus Playout VS100 Video Server Datasheet.pdf").
- [IcaP1] CD Emp. Icareus: Oferta de la solución de la casa Icareus("IcaP1 Oferta IC09T25.pdf")
- [Icn1] CD Emp. Ican-t: Especificaciones sobre receptores de la casa I-cant ("Ica1 especificaciones i-cant3810T.pdf").
- [Icn2] CD Emp. I-cant: Especificaciones sobre receptores de la casa I-cant ("Ica2 iCAN2000T\_web.pdf").
- [Inv1] CD Emp. Invest: Especificaciones sobre receptores de la casa i-cant ("Ica2 iCAN2000T\_web.pdf").
- [Inv2] CD Emp. Invest: Especificaciones sobre receptores de la casa i-cant ("Inv2 especificaciones inves-TDT-MHP-2300M.pdf").

- [Inv3] CD Emp. Invest: Especificaciones sobre receptores de la casa i-cant ("Inv3 Especificaciones inves-TDT-MHP-2200M.pdf").
- [ISO 13818] Cd Memoria Documentacion/iso13818-1.pdf
- [kra1] CD Emp. kramer: Especificaciones de la matriz de conmutación de Kramer ("Kra1 Kramer VS-41 HD.pdf").
- [kraP1] CD Emp. kramer: Oferta de la matriz de conmutación de la casa Kramer ("KraP1 Oferta Kramer Selector SDI con control RS-232.eml").
- [Mai1] <http://www.mainconcept.com/site/prosumer-products-4/mainconcept-online-store-21960.html>
- [MaiP1] CD Emp. Mainconcept: Oferta para el software Mainconcept Reference ("MaiP1 Oferta Mainconcept Reference.pdf").
- [Mai1] [www.mhp.org](http://www.mhp.org)
- [Mhp1] Información en Wikipedia sobre MHP  
[http://es.wikipedia.org/wiki/Multimedia\\_Home\\_Platform](http://es.wikipedia.org/wiki/Multimedia_Home_Platform)
- [Min1] [www.televisiandigital.es](http://www.televisiandigital.es), web del Ministerio de Industria Turismo y Comercio para dar información sobre la televisión digital y el apagón analógico.
- [Mit1] CD Emp. MIT-eXperts: Especificaciones para producto de la casa MIT-eXperts ("MIT eXperts iDesigner.pdf").
- [Mit2] CD Emp. MIT-eXperts: Especificaciones para producto de la casa MIT-eXperts ("MIT iMUX.pdf").
- [MitP1] CD Emp. MIT-eXperts: Oferta de la solución de la casa MIT-eXperts ("MitP1 Oferta MIT xperts iMUX.pdf").
- [Mpc1] <http://mpc-hc.sourceforge.net/about-homepage.html>.
- [Mts1] Información en Wikipedia sobre *Transport stream*  
[http://es.wikipedia.org/wiki/Transport\\_Stream](http://es.wikipedia.org/wiki/Transport_Stream)
- [Nit1] Cd Memoria Documentacion/ETSI-EN-300-468.pdf 5.2.1
- [Omb1] CD Emp. OMB: Especificaciones del modulador de la casa OMB ("Omb1 Manual MOT 5 DVB-T COMPACT 2v0 \_esp\_.pdf").
- [OmbP1] CD Emp. OMB: Oferta del modulador de la casa OMB ("OmbP1 Oferta MOT 5 DVB-T ECO.pdf").
- [OmbP2] CD Emp. OMB: Oferta del modulador de la casa OMB ("OmbP2 Oferta Modulador DVB-T.pdf").

- [Opc1] Página web de Avalpa <http://www.avalpa.com/>
- [Opc2] Cd Memoria Documentacion/*Scripts script.py*.
- [Opc3] Cd Memoria Documentacion/*Scripts script.sh*.
- [Osm1] [www.osmosys.tv](http://www.osmosys.tv)
- [Pat1] Cd Memoria Documentacion/iso13818-1.pdf 2.4.4.3
- [Peb1] CD Emp. Pebble beac Especificaciones del software de continuidad de la casa Pebble Beach ("Peb1 especificaciones Neptune Lite.pdf").
- [Pes1] Información en Wikipedia sobre PES  
[http://en.wikipedia.org/wiki/Packetized\\_Elementary\\_Stream](http://en.wikipedia.org/wiki/Packetized_Elementary_Stream).
- [Pcr1] Información en Wikipedia sobre PCR  
[http://en.wikipedia.org/wiki/MPEG\\_transport\\_stream#PCR](http://en.wikipedia.org/wiki/MPEG_transport_stream#PCR).
- [Pmt1] Cd Memoria Documentacion/iso13818-1.pdf 2.4.4.8
- [Pro1] CD Emp. Promax: Especificaciones de los productos de la casa Promax("Pro1 Promax DT-101-102 DVB-T Modulator.pdf").
- [Pro2] CD Emp. Promax: Especificaciones de los productos de la casa Promax("Pro2 Promax DT-301-302 Receptor Satellite to TS-ASI.pdf").
- [Pro3] CD Emp. Promax: Especificaciones de los productos de la casa Promax ("Pro3 Promax DT-504 Codificador V-A Analogico a COFDM.pdf").
- [Pro4] CD Emp. Promax: Especificaciones de los productos de la casa Promax ("Pro4 Promax MO-160 DVB-T Modulator.pdf").
- [Pro5] CD Emp. Promax: Especificaciones de los productos de la casa Promax ("Pro5 Promax MO-180 DVB-T modulator.pdf").
- [Pro6] CD Emp. Promax: Especificaciones de los productos de la casa Promax ("Pro6 Promax DTTV DigitalToTV.pdf").
- [Pro7] CD Emp. Promax: Especificaciones de los productos de la casa Promax ("Pro7 Promax DT-800 unidad de alimentacion.pdf").
- [Pro8] CD Emp. Promax: Especificaciones de los productos de la casa Promax ("Pro8 Promax DT-900 Estructura de pared.pdf").
- [Pro9] CD Emp. Promax: Especificaciones de los productos de la casa Promax ("Pro9 Promax DT-202 transmodulador DVB-S a DVB-T.pdf").



- [ProP1] CD Emp. Promax: Oferta de los productos de la casa Promax ("ProP1 Oferta 2009 09 24.PDF").
- [ProP2] CD Emp. Promax: Oferta de los productos de la casa Promax("ProP2 ListaPrecios Equipamiento para ICT y TDT.pdf").
- [Pts1] Información en Wikipedia sobre PTS  
[http://en.wikipedia.org/wiki/Presentation\\_time\\_stamps](http://en.wikipedia.org/wiki/Presentation_time_stamps).
- [Sap1] CD Emp. SAPEC: Especificaciones de los productos de la casa SAPEC ("Sap1 Especificaciones Multiplexor MTS1000\_DS.pdf").
- [Sap2] CD Emp. SAPEC: Especificaciones de los productos de la casa SAPEC ("Sap2 Especificaciones Encoder SIVAC3000\_MV\_DS.pdf").
- [SapP1] CD Emp. SAPEC: Oferta de los productos de la casa SAPEC ("SapP1 Oferta CPV-0910-07MVE3010.pdf").
- [Sdi1] Recomendaciones ITU-R BT.601/5 y ITU-R BT.656/5, que conforman el núcleo de lo que de modo coloquial se conoce como estándar SDI de codificación de vídeo.
- [Sid1] CD Emp. SIDSA: Especificaciones del codificador de la casa SIDSA ("Sid1 Karina Flex Encoder.pdf").
- [Sdt1] Cd Memoria Documentacion/ETSI-EN-300-468.pdf 5.2.3
- [Str1] CD Emp. Stream Labs: Especificaciones de los productos de la casa Stream Labs ("Str1 Especificaciones TeleInfo StreamLabs.pdf").
- [Str2] CD Emp. Stream Labs: Especificaciones de los productos de la casa Stream Labs ("Str2 Manual de usuario TeleInfo.pdf").
- [Str3] CD Emp. Stream Labs: SDK del software de continuidad de la casa Stream Labs("Str3 SDK TeleInfo Software Development Kit.rar").
- [Str4] CD Emp. Stream Labs: Especificaciones de los productos de la casa Stream Labs ("Str4 Especificaciones Tarjetas StreamLabs.pdf").
- [Str5] CD Emp. Stream Labs: Especificaciones de los productos de la casa Stream Labs ("Str5 Stream wind SDI.pdf").
- [Str6] CD Emp. Stream Labs: Especificaciones de los productos de la casa Stream Labs ("Str6 Stream Wind SDI II.pdf").
- [StrP1] CD Emp. Stream Labs: Especificaciones de los productos de la casa Stream Labs ("StrP1 Oferta via mail Stream Labs.eml").
- [Stro1] CD Emp. Strong: Especificaciones sobre receptores de la casa Strong ("Str1 especificacion Strong5510\_LL\_EN.pdf").

- [Stro2] CD Emp. Strong: Especificaciones sobre receptores de la casa Strong ("Str2 especificacion Strong5501\_LL\_EN.pdf").
- [Tps1] Información en Wikipedia sobre las tablas PSI de MPEG2 [http://es.wikipedia.org/wiki/Transport\\_Stream](http://es.wikipedia.org/wiki/Transport_Stream) Información en Wikipedia sobre las tablas SI de DVB <http://es.wikipedia.org/wiki/DVB-SI>
- [Tbr1] Información en Wikipedia con tablas de tasas de bit para DVB <http://en.wikipedia.org/wiki/DVB-T>.
- [Tan1] CD Emp. Tandberg: Especificaciones de los productos de la casa Tandberg ("Tan1 Especificaciones Codificador Tandberg E5710.pdf").
- [TanP1] CD Emp. Tandberg: Oferta de los productos de la casa Tandberg ("TanP1 Oferta tan09255jh.pdf").
- [Tdt1] Cd Memoria Documentacion/ETSI-EN-300-468.pdf 5.2.5
- [Tdt1] <http://www.mityc.es/es-es/gabineteprensa/notasprensa/documents/np%20tdt%20de%20pa%20go%2013%2008%2009.pdf>
- [Tmi1] CD Emp.Tmira: Especificaciones de los productos de la casa Tmira ("Tmi1 Especificaciones TmBroadcast.pdf").
- [Tmi2] CD Emp.Tmira: Especificaciones de los productos de la casa Tmira ("Tmi2 especificacionesTmInteractive.pdf").
- [TmiP1] CD Emp.Tmira: Oferta de los productos de la casa Tmira ("TmiP1 Oferta tmBroadcast Aranguren.pdf").
- [Tvm1] CD Emp. Epsilon Media: Especificaciones del software de continuidad de la casa Visión Robotics ("Tvm1 especificaciones Plataforma TVmakerTDTSuite.pdf").
- [Tvm2] CD Emp. Epsilon Media: Especificaciones del software de continuidad de la casa Visión Robotics ("Tvm2 especificaciones Toucan 100.pdf").
- [Tvm3] CD Emp. Epsilon Media: Especificaciones del software de continuidad de la casa Visión Robotics ("Tvm3 especificaciones Flamingo660.pdf").
- [Ubu1] Página web de Ubuntu <http://www.ubuntu.com/>
- [Vec1] CD Emp. Vektor3: Especificaciones del software de continuidad de la casa Vector3 ("Vec1 Especificaciones VECTORBOX DSX.pdf").

- [Vec2] CD Emp. Vertor3: Especificaciones del software de continuidad de la casa Vector3 ("Vec2 Especificaciones VECTORBOX 6000.pdf").
- [VecP1] CD Emp. Vertor3: Oferta del software de continuidad de la casa Vector3 ("VecP1 Oferta VectorBox DSX 0710092\_DSX.pdf").
- [VecP2] CD Emp. Vertor3: Especificaciones del software de continuidad de la casa Vector3 ("VecP2 Oferta VectorBox 6000 0110092\_6000.pdf").
- [VecP3] CD Emp. Vertor3: Oferta del software de continuidad de la casa Vector3 ("VecP3 Vector3 oferta HD.eml").
- [Vlc1] Página web de Video LAN Client: <http://www.videolan.org/vlc/>.
- [Vlc2] Pagina web donde se describe el lenguaje usado para escribir páginas web dinámicas para el interfaz http del VLC [http://wiki.videolan.org/Documentation:Play\\_HowTo/Building\\_Pages\\_for\\_the\\_HTTP\\_Interface](http://wiki.videolan.org/Documentation:Play_HowTo/Building_Pages_for_the_HTTP_Interface)
- [Vlm1] Página web de Video LAN Manager <http://www.videolan.org/doc/streaming-howto/en/ch05.html>
- [Vsn1] CD Emp. VSN: Especificaciones del software de continuidad de la casa VSN ("Vsn1 VSNmatic Informacion de producto.pdf").
- [Vsn2] CD Emp. VSN: Especificaciones del software de continuidad de la casa VSN ("Vsn2 VSN Tapeless Television SD and HD.pdf").
- [VsnP1] CD Emp. VSN: Oferta del software de continuidad de la casa VSN ("VsnP1 Oferta UPNA 280909v1.xls").